UNIVERSITE Jean Monnet – SAINT-ETIENNE

Laboratoire Hubert Curien

Membre de l'Université de Lyon

ECOLE DOCTORALE Saint-Etienne Sciences, Ingénierie, Santé

UNIVERSITE Politehnica - BUCAREST

Doctorat

IMAGE/VISION/SIGNAL

Andreea SMOACĂ

ID Photograph hashing: a global approach

Thèse dirigée par ***Thierry FOURNEL***
Vasile LĂZĂRESCU

Soutenue le 12.12.2011

Jury :

Prof. Patrick Bas	Univ. Lille, France	examinateur
Prof. Jean-Marie Becker	Univ. CPE Lyon, France	examinateur
Prof. Philippe Bolon	Univ. Savoie, France	rapporteur
Prof. Dinu Colțuc	Univ. Valahia Târgoviște, Roumanie	rapporteur
Prof. Dr. Thierry Fournel	Univ. Jean Monnet, France	examinateur
Prof. Alexandru Isar	Univ. Politehnica Timişoara, Roumanie	examinateur
Prof. Vasile Lăzărescu	Univ. Politehnica Bucarest, Roumanie	examinateur
Prof. Teodor Petrescu	Univ. Politehnica Bucarest, Roumanie	président

Acknowledgments

I would like to begin by expressing my sincere gratitude to my advisor Prof. dr. ing. Vasile Lăzărescu for his guidance and support in every stage of my graduate study. I am also grateful to my other advisor, Prof. Thierry Fournel, for his guidance, nurturing, affection and encouragement. I appreciate all their contributions of time, ideas, and funding to make my PhD experience productive and stimulating.

This thesis would not have been possible without the help of Conf. dr. ing. Daniela Colţuc. I am grateful for her constant guidance, support, affection and patience during the past three years. The enthusiasm she has for her research was contagious and motivational for me, even during tough times in the Ph.D. pursuit. Hers knowledge, kindness, patience and vision have provided me with lifetime benefits.

I would like to thank my committee members, Prof. Bas, Prof. Becker, Prof. Bolon, Prof. dr. ing. Colţuc, Prof. dr. ing. Isar, Prof. dr. ing. Petrescu. (in alphabetical order). I am honored to have them serve on my committee.

I am blessed to have true friends who offered me tremendous affection and support. I am particularly grateful to my friend and colleague Dragoş for his love, encouragement, support and help during my PhD but also to my best friends Dani and Irina. I can not thank them enough for their unconditional love and moral support in the tough periods of my life.

The years spent in France would not have been as wonderful without my new best friends Anca and Mariana. Their friendship is the most precious gift I received during my PhD. They taught me how to enjoy life outside work and offered my the love and carrying of a family.

I would also like to thank my other friends from Romania and France for the wonderful friendship they offered me. I would particularly like to thank Alex and Elena for their advices, suggestions, love and moral support, but also to Andreea, Asma, Alessandro, Ionel, Laura, Lucian, Roxana, Sebastian, Sylvain, Valentina, Umile for the special moments they made me live.

I am thankful to my colleagues from the Politehnica University and Hubert Curien Laboratory who provided a very pleasant environment for quality work to flourish. Thanks to Audrey, Bogdan, Florin, Iulian, Jacques, Mădălin and Muhammad for their friendship and for sharing the glory and sadness of conferences deadlines and day-to-day research.

Thanks also to my family, specially my mother who accomplished without complaints the endless errands that I asked her to do, even when she was on the peak of stress and lack of sleep because of her job and family problems. Thanks also to my father for being the reason I have chosen a career in engineering and research and for being a role model for his daughter. I dedicate my thesis to you mom and dad. Thank you for your endless

love. I will always be grateful.

Abstract

This thesis addresses the question of the authenticity of identity photographs, part of the documents required in controlled access. Since sophisticated means of reproduction are publicly available, new methods / techniques should prevent tampering and unauthorized reproduction of the photograph.

This thesis proposes a hashing method for the authentication of the identity photographs, robust to print-and-scan. This study focuses also on the effects of digitization at hash level. The developed algorithm performs a dimension reduction, based on independent component analysis (ICA). In the learning stage, the subspace projection is obtained by applying ICA and then reduced according to an original entropic selection strategy. In the extraction stage, the coefficients obtained after projecting the identity image on the subspace are quantified and binarized to obtain the hash value.

The study reveals the effects of the scanning noise on the hash values of the identity photographs and shows that the proposed method is robust to the print-and-scan attack. The approach focusing on robust hashing of a restricted class of images (identity) differs from classical approaches that address any image.

Résume

Cette thèse traite de la question de l'authenticité des photographies d'identité, partie intégrante des documents nécessaires lors d'un contrôle d'accès. Alors que les moyens de reproduction sophistiqués sont accessibles au grand public, de nouvelles méthodes / techniques doivent empêcher toute falsification / reproduction non autorisée de la photographie d'identité.

Cette thèse propose une méthode de hachage pour l'authentification de photographies d'identité, robuste à l'impression-lecture. Ce travail met ainsi l'accent sur les effets de la numérisation au niveau de hachage. L'algorithme mis au point procède à une réduction de dimension, basée sur l'analyse en composantes indépendantes (ICA). Dans la phase d'apprentissage, le sous-espace de projection est obtenu en appliquant l'ICA puis réduit selon une stratégie de sélection entropique originale. Dans l'étape d'extraction, les coefficients obtenus après projection de l'image d'identité sur le sous-espace sont quantifiés et binarisés pour obtenir la valeur de hachage.

L'étude révèle les effets du bruit de balayage intervenant lors de la numérisation des photographies d'identité sur les valeurs de hachage et montre que la méthode proposée est robuste à l'attaque d'impression-lecture. L'approche suivie en se focalisant sur le hachage robuste d'une classe restreinte d'images (d'identité) se distingue des approches

classiques qui adressent une image quelconque.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Problem statement

Due to the increase of digital technologies, person's authentication is demanded in more and more situations such as transactions, voting, border crossing etc. Nowadays, worldwide users have access to powerful software and to high–quality technologies such as cameras, scanners and printers, which allow them to easily edit and reproduce identity document (ID). Thus the need for ownership protection and prevention of unauthorized modifications of ID documents is highly required.

Traditional methods like knowledge–based passwords or token–based cards are still employed in authentication but identity documents are preferred because various features can be inserted in order to increase security and to limit forgery. For images, new techniques for authentication such as watermarking and perceptual hashing have been proposed. Watermarking is a popular method in image/document authentication, which has made substantial progress in the past few years. It consists in embedding information, called watermark, into the content of the image, information which can be further used to verify the data's authenticity. Robust hashing is a viable alternative to digital watermarking which maps the multimedia data to a compact representation (called label, fingerprint, hash value or digital signature) of it. The main difference between the two methods is that the latter is always content–dependent.

Since most of the ID documents contain the photograph of the holder, in this thesis I focused my attention on ID image forgery and I have proposed to use perceptual image hashing as an image authentication technique. My contribution consists in implementing a robust face hashing algorithm which may be used in access control, border control, visas issuance department etc. For example, in access control, a person must first identify himself before being allowed to enter the secured area (school, concert, work places). This can be done through an access card which contains various information of the person like user's photo, name, an embedded hash value of the photo etc. The person will be allowed to enter only if he passes the security check. This consists in scanning the photo printed on the card, extract its hash value and compare it with the one embedded in the card.

1.2 Approach

Most of the existing image hashing algorithms have been developed for a wide class of images. The ID picture is a particular case of image with special features that should be taken into account when designing a hash algorithm. The proposed technique for extracting the hash value of an ID picture is a learning–based approach that takes into consideration the characteristics of the face, such as eyes, mouth, nose etc. In the learning process, the ICA is performed in order to obtain a projection subspace consisting in a series of face features. Each scanned ID photo is projected onto the learned subspace and the projection coefficients are retained. The hash value associated to the photo is the vector of the quantized and binarized coefficients.

1.3 Outline

This thesis focuses on extracting a hash value from ID photographs so that the hash value should be invariant under perceptually insignificant modifications on the image and sensitive to malicious manipulations on the image.

Chapter 2 presents a framework for image hashing. The desired properties of a perceptual hash are defined and the main challenges in finding good trade–offs between these proprieties are identified. The previous work made in image hashing is reviewed. Early methods, that achieve robustness for perceptually insignificant attacks, are usually insecure. Recently developed methods, based on dimension reduction, have shown to outperform the previous techniques.

Chapter 3 provides an introduction to the concept of ICA. ICA is presented as a case of Principal Component Analysis (PCA), which provides an independent rather than uncorrelated image decomposition. Various algorithms employed to obtain independence are described

Chapter 4 develops a scheme for ID photograph authentication based on two types of ICA representation. The first representation is more pertinent. An image is defined as a linear combination of face features rather than a linear combination of faces. Several algorithms for face features selection are further developed and can be combined to obtain a better performance.

Chapter 5 describes the print–and–scan process. The printing and scanning devices are characterized for a better understanding of the physical phenomena that occur in this complex process. The effects caused by printing and scanning an image are identified. Countermeasures to these effects and statistical models for the print–and–scan noise are also presented.

Chapter 6 develops a model for the scan noise that affects the printed ID photograph in the authentication process. The two main noises and their effects are analyzed both at image and hash level. It is shown that the noise is content–dependent and that the algorithms for face features selection proposed in chapter 5 allow to reduction of the scan noise influence on the hash.

Chapter 7 concludes the thesis by summarizing the contributions and presents directions for future work.

Chapter 2

State of art in image hashing

2.1 Introduction

This chapter presents a unifying framework of image hashing. Section 2.2 highlights the place of image hashing in the multimedia data authentication context. It also shows the two main applications of image hashing, image authentication and indexation. Next, the desired properties of a perceptual image hash are given.

Section 2.3 reviews the work in this domain and at the end the approach proposed in this thesis is briefly described. Section 2.4 summarizes the main ideas presented in this chapter.

2.2 Image hashing in multimedia data authentication and archiving

The evolution of digital technology during the past few decades has led to significant changes in the modern society. Along with powerful tools, new devices, such as high quality digital cameras, scanners and printers, have reached users worldwide. They allow to create, edit and reproduce audio sequences, images, videos and documents. In this context, the protection of ownership and the prevention of unauthorized tampering have become significant concerns [Cox 2002, Wu 2002].

In order to verify the reliability of the multimedia data, several authentication methods have been developed. A solution is the use of conventional cryptographic hashes as Message Digest 5 (MD5) [Rivest 1992] or Secure Hash Algorithm (SHA) [NIST 2008] which are deeply perceptive even to a single bit change in the input data. Thus, the data integrity can be verified when every bit is unaffected. But, in the case of multimedia, data cryptographic hashes are no longer suitable. The multimedia data can exist in different digital forms and is still considered authentic even after non–malicious attacks like JPEG compression or print–and–scan attacks in the case of images. This is the reason why hash algorithms that are content–dependent and tolerate content–preserving distortion (same or similar content lead to same hash, while different content lead to drastically changes in the hash) were developed. These hashes must take into consideration the changes in

the visual/audio domain and retain only the most significant perceptual attributes of the media. Therefore, a *multimedia hash* is defined as a content–based digital signature of the media data. In literature it is also called *label* or *fingerprint*.

Many multimedia hash functions have been developed for media authentication. They can be divided into two main categories: *unkeyed hash functions* and *keyed hash functions*. The *unkeyed hash functions* generate the multimedia hash from an arbitrary input while *keyed hash functions* need an arbitrary input and a secret key.

In media authentication, the keyed hash functions are utilized. In an authentication system like the one in Fig. 2.1, the *hash function* extracts certain characteristics from the data that are further encrypted by using a secret key. The hash is sent along with the media either by appending or embedding it to the original data. At the receiver, the hash value is computed and confronted with the one transmitted along with the data, in order to verify its authenticity.

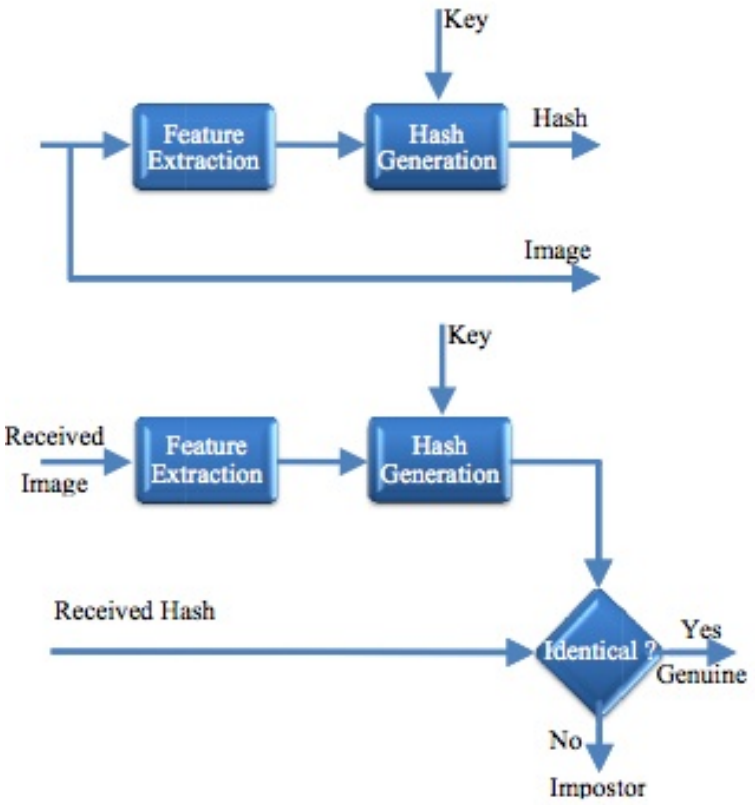


Figure 2.1: Image authentication with keyed hash function.

Beside digital signature–based algorithms [Venkatesan 2000, Mihçak 2001] [Kailasanathan 2001, Martinian 2002], another method widely used for media authentication is watermarking [Cox 1997, Lin 1999b, Yeung 1997, Wu 1998, Wolfgang 1999, Xie 2001]. Watermarking is defined as the process of inserting invisible (or inaudible) data into the media (Fig. 2.2). In authentication applications, the embedded watermark

is retrieved at the receiver side in order to take a decision on the authenticity of the media.

The major difference between a watermark and a hash is that the latter is content–dependent while the former is not. In the first case, the watermark integrity is verified, while in the second case it is the image content. Nevertheless, both the watermark–based approach and the digital signature–based approach for media content authentication are required to be perceptive to any malicious attack such as attacks against the key, intentional replacement of pixel values, forging the original image in order to claim multiple identities. At the same time they should remain robust to non–intentional modifications such as JPEG compression (with compression ratios that do not alter significantly the perceptual quality of the media) or print–and–scan attacks.

Figure 2.2: Media authentication methods: a) image hashing; b) invisible watermarking.

Another important application for multimedia hashing is content–based retrieval from databases [Lin 2001]. Simple methods for image search in databases such as sample–by–sample comparisons are computationally inefficient. Because these methods compare the lowest level of content representation they are not robust to attacks such as geometric distortions. Robust image hash functions can be used to address this problem [Venkatesan 2000]. For each image entry in the database a hash is computed and stored with the original image in the form of a lookup table. To search for a given image in the database, its hash is computed and compared with the hashes in the lookup table. The data entry corresponding to the closest match, in terms of certain hash–domain distance, is then fetched. Since the hash has a much smaller size than the original image, matching the hash values is computationally more efficient than the sample–by–sample comparison.

Image hash functions have also been employed in some other applications such as image and video watermarking and information embedding. In image watermarking, instead of inserting a mark which is independent of the content of the image, the hash of the image can be computed and embedded (Fig. 2.3). The hash functions can also be employed as image–dependent keys for watermarking [Holliman 1999]. In video watermarking, if the attacker has in possession multiple copies of similar frames, he can deduce the watermark by using "collusion attacks" [Su 2005]. A proposed solution to this problem is to generate the watermark using a secure, content–dependent hash value as a

Figure 2.3: Watermarking with image hash embedding.

key generator [Fridrich 2000].

When designing a robust image hash algorithm, the following constraints must be taken into consideration:

- *robustness* refers to the ability of the hash to produce equivalent hashes for input images that differ by a certain distortion level. An acceptable distortion level should take into account some non–malicious attacks;

- *security* consists in introducing a secret key when generating the hash value. Without the knowledge of this key the hash values cannot be forged;

In indexing applications, additional constraints must be taken into consideration. For example, the hash length has to be as small as possible in order to guarantee fast search in the hash database and in the same time, the hash length has to be as large as possible to satisfy scalability requirements.

2.2.1 A typical scheme for robust image hashing

In order to attain robustness and security, almost all hashing schemes are three–step processes. A typical hashing scheme is presented in Fig. 2.4. First, perceptually significant features are extracted from the content, then quantized and compressed to a short binary string. For content authentication, the hash is encrypted.

In practice, the hash value does not remain exactly the same, it slightly changes because of non–malicious attacks. Therefore, the hash comparison is usually a hypothesis testing problem: a similarity measure between the received hash and the computed one is calculated and compared with a threshold; if it is below the threshold, the tested version is considered as authentic, otherwise as inauthentic.

Figure 2.4: A typically hashing scheme.

2.2.2 The properties of an ideal hash

Let I be the image to be hashed. The hash function takes the image I and a secret key K in order to produce an n–bit hash $h = H(I, K)$.

An ideal hash must have the following properties:

1) *Perceptual robustness*: for any pair of two perceptually similar images, the hashes should map to the same value (Fig. 2.5).

$$P(H(I, K) = H(I_{ident}, K)) = 1 \tag{2.1}$$

where P denotes the probability, and I and I_{ident}, the original, respectively the perceptually identical image.

(a) Original image (b) JPEG compression image

Figure 2.5: Two perceptually identical images.

2) *Sensitivity*: two perceptual distinct images (images of different persons or original and tampered image) must lead to different hash values (Fig. 2.6).

$$P(H(I, K) \neq H(I_{dif}, K)) = 1 \tag{2.2}$$

where I and I_{dif}, represent the original, respectively the different/tampered image.

(a) Original image (b) Tampered image (c) Different person

Figure 2.6: a) Original and tampered image; b) Different image.

3) *Unpredictability of the hash* consists in having an approximatively uniform distribution of the hash

$$P(H(I,K)=v) \approx \frac{1}{2^q}, v \in \{0,1\}^q \tag{2.3}$$

where v is the hash value extracted from I and $q \in N$.

There is a trade–off between robustness and sensitivity. The first property requires robustness under small distortions while the second imposes minimal collision probability for different hashes. The last two properties are important from the security point of view, i.e. it should be very difficult for an attacker to forge the content of an image and obtain the same hash value as for the original image.

2.3 Review of image hashing algorithms

This section reviews the research which has been done so far in image hashing. The approaches can be classified into methods based on image statistics [Schneider 1996, Kailasanathan 2001, Venkatesan 2000], pixel/coefficient relations [Lin 1997, Lu 2000], preservation of coarse image representation [Fridrich 2000, Mihçak 2001] low–level image feature extraction [Bhattacharjee 1998, Dittmann 1999] and dimension reduction techiniques [Kozat 2004].

2.3.1 Image Statistics Based Approaches

In one of the first approaches, Schneider *et al.* [Schneider 1996] use intensity histograms of image blocks for media authentication. The authentication process consists in computing the Euclidean distance between the histogram of the authentic image and the candidate image. The authors use as authenticity measure the sum of all Euclidean distances over the image.

In [Kailasanathan 2001], three methods for image authentication that survive to content adjustments by JPEG compression have been described.

Method 1

This method uses mean and standard deviation for generating the hash value. The statistical values that capture the significant features of an image and remain basically unchanged through JPEG compression are obtained from the original image I_o and all the JPEG compressed versions of image I_o (down to a tolerated quality factor σ), denoted I_m.

The hash vector is generated as follows:

Step 1 The image I_o and each JPEG compressed version I_m are divided into m x n rectangular blocks of size a x b;

Step 2 For each square block, the mean/standard deviations S_{ij}^o and S_{ij}^m ($i = 1, ..., m$ and $j = 1, ..., n$) are computed;

Step 3 A threshold T_σ is computed :

$$T_\sigma = \max_{i,j,m} |S_{ij}^o - S_{ij}^m| \tag{2.4}$$

The final hash vector contains the series of statistics of the original blocks S_{ij}^o, the block dimension a x b, the type of statistics utilized and the threshold T_σ. The length of the hash vector will be given by the block size.

In the verification stage, for a candidate image I_c, the statistics S_{ij}^c are computed by performing steps 1) and 2) . The candidate image I_c is considered authentic if $|S_{ij}^o - S_{ij}^c| \leq T_\sigma$.

Method 2

In this method, instead of the fixed sized blocks, the authors use images obtained by k–means clustering for hash generation and verification.

The steps in generating the hash vector can be summarized as follows:

Step 1 The image I_o and each modified version I_m are divided into P regions by using the k–means algorithm. The pixel features used by k–means are the pixel value, the mean and standard deviation of adjacent pixels [Kailasanathan 2001]. The code book provided by k–means is included in the hash value;

Step 2 For each region $i = 1, ..., P$, S_i^o and S_i^m are computed;

Step 3 The threshold T is determined based on the value D_m as in equation 2.6. The value D_m of the Euclidean distance between the standard deviation of the original image regions S_i^o and the modified image regions S_i^m is:

$$D_m = \sqrt{\sum_{i=1}^{P} ((S_i^o) - (S_i^m))^2} \tag{2.5}$$

$$T = max_m(D_m) \tag{2.6}$$

The hash consists of the sequence of original image regions statistics S_i^o, the number of regions P, the type of statistics, the code book and the threshold T.

In the verification stage, the candidate image I_c is divided into P regions by using k–means algorithm and the code book. Then, the standard deviations S_i^c of these regions are computed. The candidate image is considered authentic if the difference $D_c \leq T$.

Method 3

In this method, the P regions are obtained by vector quantization with Linde–Buzo–Gray (LBG) algorithm.The feature vectors are the same as in method 2 i.e. the pixel value, mean of adjacent pixels and standard deviation of adjacent pixels.

The hash generation can be resumed as follows:

Step 1 : Take the original image I_o and obtain m images by compressing it for m different JPEG quality factors;

Step 2 : For all the above images the feature vectors T_{c_i} are obtained. The training set from the LBG algorithm consists in $T_c = T_{c_1} U T_{c_2} U ... U T_{c_m}$;

Step 3 Train the codebook on the T_c obtained only from the compressed images feature vectors. Let $FC = \{C_1, C_2, ..., C_p\}$ be the codebook and p is the size of the codebook;

Step 4 : Starting with the codebook FC created above make one more iteration on LBG algorithm using as training set T_{c_i}. Let $FC' = \{C_1', C_2', ..., C_p'\}$ be the codebook after one iteration;

Step 5 : Determine a range *[a, b]* of acceptable Euclidean distances by looking at the minimum and maximum values of ED_{comp}.

The hash is obtained by concatenating *FC*, *a* and *b*. The length of the hash is ((p x 3)+2)x r bytes, where r is the number of bytes employed to represent a real number.

In the verification stage, for a candidate image I_c,the sequence of the feature vectors is computed. Assuming FC^r is the codebook obtained after one more iteration of LBG algorithm, the deviation caused in Euclidean distance is determined as $ED_{received} = \sqrt{\sum_{i=1}^{p} (C_i - C_i^r)^2}$. The candidate image is considered authentic if $ED_{received} \in [a, b]$.

In [Venkatesan 2000], a hash algorithm based on image statistics extracted from different Wavelet Transform (WT) sub–bands is proposed. It is a keyed–hash algorithm that gets as inputs the image to be hashed and a secret key K. The key K is employed as a seed for the pseudorandom number generator employed during several stages of the proposed method.

The main steps of Venkatesan *et al.* algorithms can be summarized as follows:

Step 1 *Random tiling transform by using K and statistics calculation*: after image wavelet decomposition, each sub–band is randomly tiled into smaller rectangles

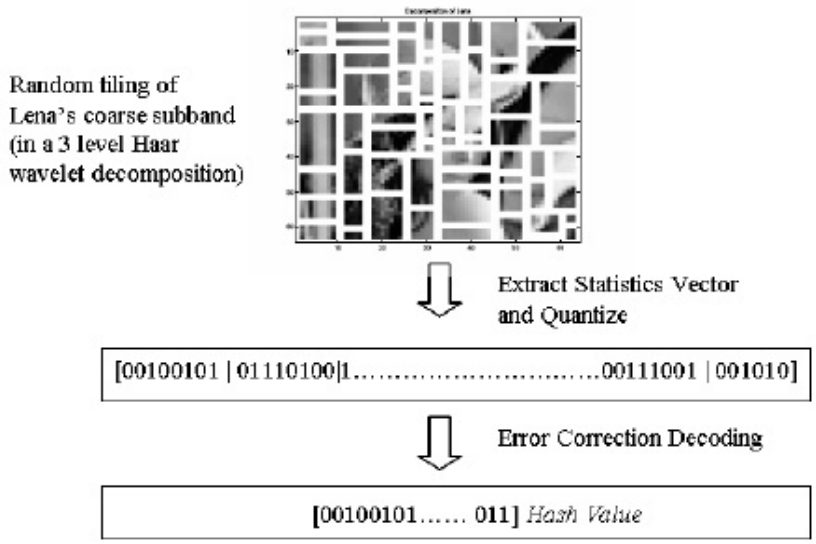


Figure 2.7: Venkatesan hash algorithm

(Fig. 2.7). Next, an image statistics vector m is computed. m is a vector consisting in the mean of coarse sub–band coefficients and the variances of the other sub–bands coefficients;

Step 2 *Randomized rounding*: in this stage the robustness against attacks is increased by quantizing the vector m. The authors consider that the inputs are generated by unknown sources and utilize randomized rounding as an essential source of randomness. The key K from the previous step is used here to obtain the distribution used for quantization. The length of quantized vector x will be three times longer since each value is represented on three bits;

Step 3 *Reed–Muler decoding*: decoding is necessary for stabilizing the hash value. A first–order Reed–Muller error-correcting decoder is employed to obtain the binary hash from the binary vector x [Blahut 1994]. The exponential pseudo norm metric is used in the decoding scheme;

Step 4 *Supplementary decoding*: The final hash value may be mapped to a shorter string;

In the verification stage, the hash value of a candidate image I_c, computed as above, is compared with the hash value of the original image by using the normalized Hamming distance.

2.3.2 Image Relations Based Approaches

Relation–based methods consist in constituting suitable image hash values based on a transform domain representation of the image, such as Discrete Cosine Transform (DCT) or WT. In these approaches, invariant relationships between the transform coefficients are identified. Lin and Chang report in [Lin 1997] a relation–based approach for image

authentication, tolerant to JPEG compression. They take into consideration the relationship between any two DCT coefficients placed at the same position in two distinct 8 x 8 blocks.

In JPEG compression, the DCT coefficients quantization and rounding are lossy operations. The authors propose a scheme for generating robust feature vectors that are not affected by these lossy operations. They prove that robust feature vectors can be obtained if all the following properties are valid:

a) if $\Delta F_{p,q}(v) > 0$, then $\Delta \bar{F}_{p,q}(v) > 0$;

b) else if $\Delta F_{p,q} < 0$, then $\Delta \bar{F}_{p,q}(v) \leq 0$;

c) else $\Delta F_{p,q} = 0$, then $\Delta \bar{F}_{p,q}(v) = 0$;

where $\Delta F_{p,q} = F_p - F_q$ and $\Delta \bar{F}_{p,q} = \bar{F}_p - \bar{F}_q$. F_p and F_q, respectively $\bar{F}_p$ and $\bar{F}_q$, are the DCT coefficients of two 8 x 8 blocks of the original image, respectively their quantized approximation.

In other words, the relationship between F_p and F_q is affected by the quantization process only when, because of rounding, *greater than* or *less than* relationships become *equal*.

As illustrated in Fig. 2.8 the hash is generated by encrypting the extracted feature vectors Z. The feature extraction process consists in generating sets of feature codes from N DCT blocks. Then the difference $\Delta F_{p,q}$ from the 8 x 8 DCT blocks is computed. If $\Delta F_{p,q} \leq k$, where k is a user–defined threshold, a feature cod bit of "0" is added, else a "1" is added;

In the encryption stage, in order to enforce robustness, some additional information is added (the mean of DCT coefficients in each 8 x 8 block position) for obtaining the hash S.

The authentication process is a three–parts process. First, DCT is performed on the received (raw or JPEG compressed) image. In the case of the compressed image, an entropy decoder and de–quantizer are also used. Secondly, the hash vector S is decrypted and then sent to the authentication comparator. The comparator will decide if the image is authentic or not by checking the mean values in each position of the DCT coefficients in the original, respectively the tested image.

Even if this approach is robust to JPEG compression, it is still sensitive to other perceptually minor modifications like cropping and shifting.

In multi–resolution wavelet analysis a certain number of parent–child pairs $< p, c >$ exists. Lu *et al.* represent them in [Lu 2000] under the form of an image called structural digital signature (SDS). The generation of the SDS is done by observing the inter–scale relations of the image wavelet coefficients. The proposed method is based on the following assumptions:

- the inter–scale relationships of the Discrete Wavelet Transform (DWT) coefficients should be preserved if content–preserving manipulations are made on the image;

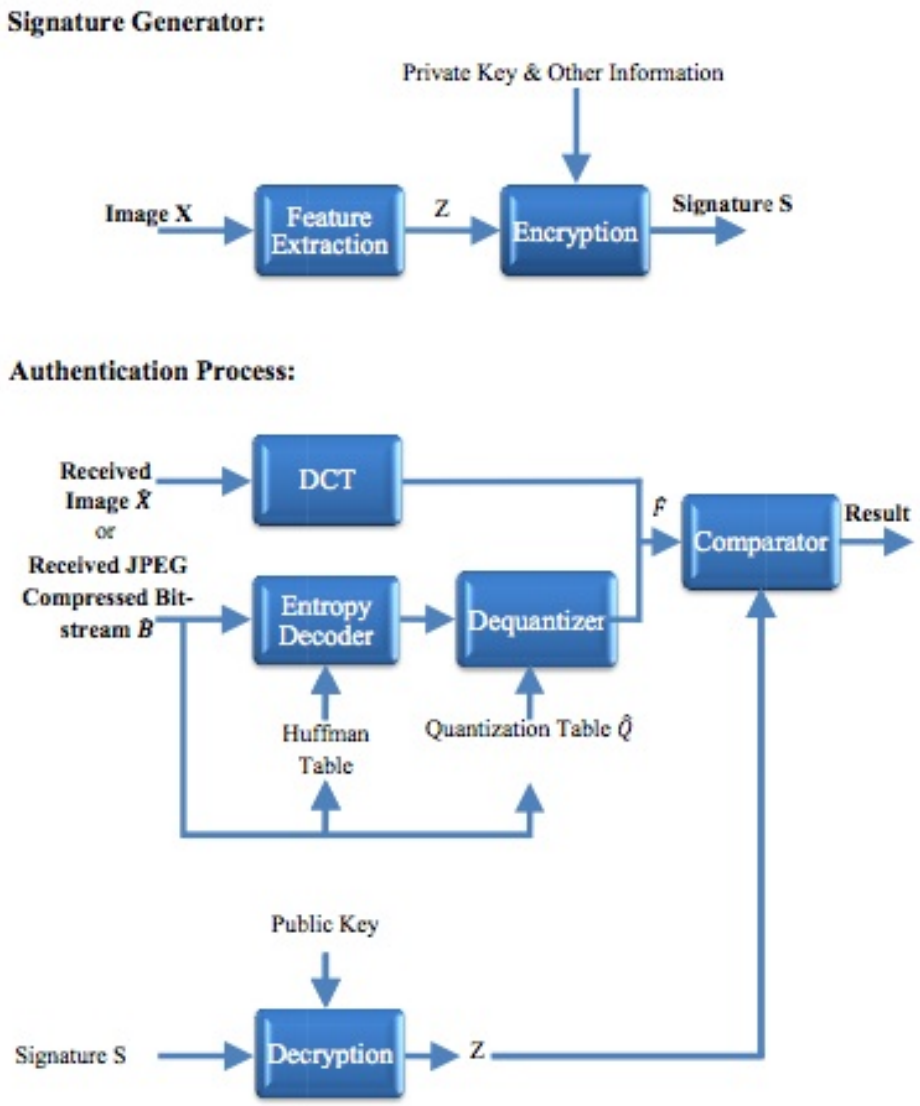


Figure 2.8: Media authentication scheme proposed by Lin and Chang [Lin 1997].

- this inter–scale relationship should be difficult to be preserved after content–changing manipulations.

Let $w_{s,o}(x, y)$ be a wavelet coefficient at scale s, orientation o and position (x, y) of image I. The SDS of image I contains a sequence of parent–child pairs that satisfies:

$$||w_{s+1,o}(x, y)| - |w_{s,o}(2x + i, 2y + j)|| \geq \sigma \ (\sigma \geq 0) \tag{2.7}$$

where σ determines the number of parent–child pairs. The greater σ is, the smaller the number of pairs in the SDS. Depending on the type of relationship the pairs carry, a symbol is assigned to each of them. The above considered symbols and their position in the wavelet domain are coded by a public key algorithm such as the Rivest, Shamir,

Adleman (RSA) method [Menezes 1996], stored and employed later in image authentication.

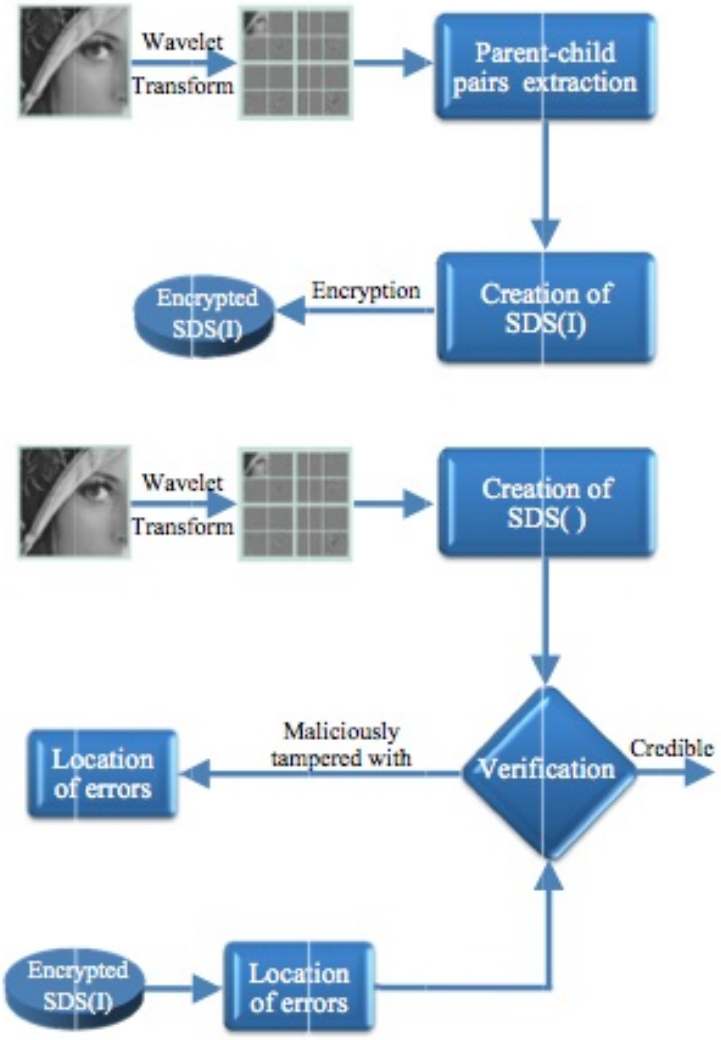


Figure 2.9: Media authentication scheme proposed by Lu *et al.* [Lu 2000].

In the authentication process (decribed in Fig. 2.9), the similarity degree between the original I and the candidate $\bar{I}$ image is defined as:

$$SIM(SDS(I), SDS(\bar{I})) = \frac{N^+ - N^-}{|SDS(I)|} \tag{2.8}$$

where $|SDS(I)|$ denotes the number of parent–child pairs, N^+ the number of pairs which satisfy 2.9 and N^- the number of pairs which do not satisfy 2.9.

$$sym(< p, c >) = sym(< \bar{p}, \bar{c} >) \tag{2.9}$$

where $sym(< p, c >)$ represents the signature symbol of the parent–child pair and $< \bar{p}, \bar{c} >$ in $\bar{I}$ is the corresponding pair of $< p, c >$ in I. The signature symbol is extracted from

the DCT pairs whter p is larger than c or not. If the $SIM(SDS(I), SDS(\bar{I}))$ value is greater than a threshold, the image is considered authentic. In the case of incidental attacks, the SDS of the candidate image is not significantly affected while in the case of malicious attacks it will be significantly destroyed.

2.3.3 Approaches based on coarse image representations

In [Fridrich 2000] a robust, key–dependent, hash function for digital watermarking is proposed. The main idea of the method is derived from the observation that large values of low–frequency DCT coefficients will automatically lead to significant perceptual changes to the image.

The procedure of hash generation is randomized by replacing the DCT modes with random smooth patterns made DC–free by mean extraction. The random smooth patterns P are obtained by applying a low–pass filter to N sub–unitary random matrices, generated by using a secret key K. Each image I is divided into image blocks of 64x64 pixels. The image blocks B are then projected on the space produced by the N patterns P. Their absolute value is computed and compared with the threshold Th in order to obtain N bits b_i:

$$\begin{aligned} &if\ |B \cdot P^{(i)} \leq Th| \quad then \quad b_i = 0 \\ &if\ |B \cdot P^{(i)} \geq Th| \quad then \quad b_i = 1 \end{aligned} \tag{2.10}$$

where $i = 1, ..., N$.

The projections are independent of the mean gray value of the block, but are image–dependent. The N bits sequence order should be adjusted so that half of the b_i bits have zero value and the other half one value, in order to provide maximum information.

The hash value contains the N bits extracted as presented above. The method is resistant to attacks like JPEG compression, additive uniform noise, sharpening, blurring, median, and mosaic filtering etc., but sensitive to geometric attacks, such as rotation, shifingt and change of scale.

Mihcak proposes in [Mihçak 2001] a two–step algorithm for obtaining the hash of an image:

- generation of intermediate hash value;
- generation of final hash value by applying a randomized lattice vector quantization to obtain the final hash vector;

The authors propose two algorithms:

Algorithm A

In this approach the significant regions of the discrete WT are retained. An iterative filtering technique is employed in order to minimize the geometrically weak components and improve the geometrically strong components. The filtering is based on region growing. This means that the blocks with isolated important components (geometrically weak)

are viable candidates to be erased via modifications, while blocks which have massive clusters of important components (geometrically strong) would probably not be erased but only a bit perturbed.

Here are the steps for generating the hash:

Step 1 For a given image I, find the WT up to level L. Let I_A be the resulting approximation sub–band.

Step 2 A binary map M is obtained from I_A by performing the following thresholding operation:

$$M(i,j) = \begin{cases} 1 & \text{if } I_A \geq T \\ 0 & \text{otherwise} \end{cases}$$

where T is selected so that the normalized Hamming weight of the binary input M should be approximatively q, where q is a sub–unitary number;

Step 3 Geometric region growing. Let $M_1 = M$ and $ctr = 1$;

Step 3.1. Determine M_2 by applying an order-statistics filter on M_1;

Step 3.2. Let $M_3 = AM_2$, where A is an algorithm parameter. M_4 is obtained by a 2–dimensional linear shift–invariant filtering on M_3.

Step 3.3. A binary map M_5 is obtained by performing a thresholding operation on M_4 similar to the one described in step 2.

Step 3.4. The geometric region growing stops if $ctr \geq C$. If it is not the case, increment ctr and restart geometric region growing for $M_1 = M_5$.

Step 4 The image hash M_5.

In this approach, there is no pseudo–randomness, because no key is used. Since randomness is important for security, *Algorithm B* was developed.

Algorithm B

In this approach, the authors apply *Algorithm A* on randomly chosen blocks of the image or its wavelet transform. Random rectangles are utilized for this purpose.

Hash generation:

Step 1 Divide image I in N rectangles R_i of size w_i x h_i. Let I_i be the sub–image obtained by overlapping rectangle R_i on image I, $i = 1, ..., N$;

Step 2 Obtain a hash matrix $H(I_i)$ of I_i by applying *Algorithm A*;

Step 3 Each hash matrix is converted into a vector by randomly ordering the matrix's elements. Concatenate all the vectors in order to obtain a hash;

Step 4 The final hash is obtained by randomly selecting $\hat{M}$ bits from the vector obtained in the previous step;

In the verification stage, the normalized Hamming distance is used to compare the hashes of the original and candidate images. The candidate image is authentic if the normalized Hamming distance is smaller than a predefined threshold.

The two algorithms proved to be robust to non–malicious geometric attacks except large rotation and cropping.

2.3.4 Low–level Image Feature Extraction

In [Bhattacharjee 1998] Bhattacharjee *et al.* propose a method that extracts features inherent to the image.

The hash generation is a two–step procedure as in common cryptographic authentication schemes:

- *Feature extraction*: the feature points are extracted by using Mexican Hat wavelets which have been shown to be more stable and also less complex than other filters. The features points are given by their coordinates;
- *encryption*: the sequence of feature points is encrypted by using a public key algorithm such as RSA.

The feature points extraction has three steps:

a) The feature detection function is defined as:

$$P_{ij}(\vec{x}) = |M_i(\vec{x}) - \lambda \cdot M_j(\vec{x})| \tag{2.11}$$

where $M_i(\vec{x})$ and $M_j(\vec{x})$ are the responses of the Mexican Hat wavelet at image position $\vec{x}$ for scales i and j.

b) The set of potential feature points is given by the local maxima of function P_{ij}. The authors use a circular neighborhood with a 5 pixels radius for obtaining the local maxima;

c) A point of local maxima is accepted as feature point if the variance of the pixels situated in the point neighborhood has a value higher than a predefined threshold. A 7 x 7 neighborhood area is used for local variance computation. The threshold value is 10;

In the verification stage for a candidate image I_c the set of feature points S_c is extracted and compared with the set of feature points of the original image,S_o. I_c is considered authentic if each feature point location obtained from S_o, is present in S_c and if no feature location present in S_o is absent in S_c.

In [Dittmann 1999], Dittmann and colleagues develop a feature extraction scheme based on the edge characteristics of an image. The edge characteristics are a good indicator of image content, allowing the recognition of the object form and the homogeneity of the image.

The edge–based hash of an image I is obtained in [Dittmann 1999] as follows:

Step 1 : The edge characteristic of I is extracted by using the Canny edge detector;

Step 2 : A binary edge pattern is obtained from the edge characteristic;

Step 3 : The feature code is obtained by applying a Variable Length Code (VLC) to the binary edge pattern;

Step 4 : The hash is obtained from the feature code;

In the authentication process for a candidate image I_c the edge pattern is computed from the extracted edge characteristic and compared with the one extracted from the original image.

The approach proved to be robust to significant content–preserving attacks like transmission errors, noise, data storage errors, brightening reduction, resolution reduction,γ distortion, changes of hue and saturation, but it is sensitive to several perceptually insignificant changes like compression, quantization and scaling.

Monga *et al.* propose in [Monga 2006] an algorithm using feature extraction based on wavelets. They develop their own feature detector based on wavelets and an adaptive quantization scheme which is based on the distribution of the extracted features. They give two versions of the algorithm, a deterministic algorithm and a randomized algorithm.

The proposed feature detection method consists in three steps. First, the wavelet transform for each image location is computed. The wavelet transform is based on end–stopped wavelets. Second, the significant features are selected by looking for a local maxima of the magnitude of the wavelet coefficients in a neighborhood. Next, a pre–defined threshold is used to eliminate fake local maxima in featureless zones of the image.

Deterministic Algorithm

The authors adapt the algorithm presented in [Mihçak 2001] to lock onto a set of feature–points conserved in visually similar images.

The intermediate hash is obtained as follows:

Step 1 : For an image I, extract the feature vector f by using the proposed feature detector described above;

Step 2 : Quantize f in order to obtain the binary vector b_f^1;

Step 3 : Perform order–statistics filtering on I in order to eliminate important components. Let I_{OS} be the filtering result;

Step 4 : Perform low–pass filtering on I_{OS} in order to retain strong components. Let I_{lp} be the result;

Step 5 : Repeat steps [1] and [2] for I_{lp} in order to get b_f^2;

Step 6 : If the Hamming distance between b_f^1 and b_f^2 is smaller than a pre–defined value, the hash takes the value of the variable b_f^2, otherwise the algorithm is reapplied for $I = I_{lp}$;

Randomized Algorithm

The first algorithm proposed in [Monga 2006] does not use a secret key. For improving the security and also scalability of the hash algorithm a randomized version of the algorithm is described below. The approach makes use of a secret key K as a seed for a pseudo–random generator employed in different stages of the algorithm.

Step 1 : Random Partitioning: divide the image into overlapping circular/elliptical regions with randomly selected radii. The N regions are labeled as C_i, i = 1,...,N;

Step 2 : Rectangularization: each C_i is approximated by random rectangles, R_i;

Step 3 : Feature Extraction: extract the binary string b_i by applying the deterministic algorithm on all R_i and then concatenate all binary strings into a binary vector b;

Step 4 : Randomized Subspace Projection: randomly choose distinct indices $i_1, i_2, ..., i_A$ so that each $i_m \in [1, B]$, m = 1, 2, ..., A. A and B represent the desired length of the hash and, respectively, the hash vector length after step [3];

Step 5 : The intermediate hash is $h(I, K) = \{b_{i_1}, b_{i_2}, ..., b_{i_A}\}$

In order to detect if a candidate image is authentic, the hash value from the original and the candidate image are compared by means of Hamming distance. The algorithm proved to be robust to visually insignificant attacks like JPEG compression, contrast enhancement, Gaussian smoothing, rotation with an angle smaller than 5^o, scaling, shearing, cropping less than 20% and sensitive to content–based attacks like significant changes in image geometry.

2.3.5 Other approaches

2.3.5.1 Dimension reduction techniques

Recently, various dimension reduction–based hashing techniques have been developed. The hashing algorithm proposed in [Kozat 2004] is a two–step algorithm. First, the intermediate features are derived from pseudo–random (PR) regions via matrix invariants such as Singular Value Decomposition (SVD). Next, a PR secondary image is constructed from the intermediate features, which is further employed for extracting the final hash vector of the image.

Let I be the input image of dimension n x n. The generic hashing scheme consists in the following steps:

Step 1 : From I generate p PR overlapping rectangles A_i of dimension m x m, $i = 1, ..., p$;

Step 2 : From each rectangle A_i generate the feature vector $\vec{g_i}$. $\vec{g_i} = T_1(A_i)$, where T_1 is one of the following transforms: SVD, DCT, DWT;

Step 3 : Build a secondary image J from a PR combination of intermediate feature vectors $\{\vec{g_1}, ..., \vec{g_p}\}$

Step 4 : From image J generate r PR overlapping rectangles B_i of dimension d x d, $i = 1, ..., r$;

Step 5 : From each rectangle B_i generate a final feature vector f_i. $f_i = T_2(B_i)$, where T_2 is the SVD transform;

Step 6 : Combine the final feature vectors f_i in order to obtain the final hash vector;

The performance of the algorithm is influenced by the choice of transformations T_1 and T_2. The hash methods obtained by combining T_1 and T_2 proved to be robust under severe geometric distortions.

The results obtained in [Kozat 2004] motivated researchers to find alternative solutions in this direction. Monga and colleagues propose in [Monga 2007] a hashing algorithm based on a dimension reduction technique developed by Lee *et al.* called nonnegative matrix factorization (NMF) [Lee 2001].

Using NMF algorithm, a matrix V of size m x n is factorized in the following way:

$$V \approx WH \tag{2.12}$$

where $W \in R^{mxr}$ and $H \in R^{rxn}$.

The NMF–NMF hashing algorithm proposed by Monga *et al.* has similar structure with the one proposed in [Kozat 2004]. The steps of the algorithm are summarized below:

Step 1 : For a given image I, generate p subimages A_i with dimension m x m, $i = 1, ..., p$;

Step 2 : For each subimage A_i obtain the matrices W_i and F_i by applying NMF:

$$A_i \approx W_i F_i^T \tag{2.13}$$

Step 3 : By pseudo–randomly combining matrices W_i and F_i a secondary image J is constructed ;

Step 4 : Reapply NMF on image J in order to obtain matrices W and H:

$$J \approx WH \tag{2.14}$$

Step 5 : The columns of matrix W and the rows of matrix H are concatenated in order to obtain the hash vector;

All the steps in the algorithm are randomized by using a secret key.

A second hash algorithm, called NMF–NMF–SQ, is proposed in [Monga 2007]. The algorithm, based on obtaining pseudorandom statistics of the NMF–NMF hash, can be summarized as follows:

Step 1 : For an image I obtain the hash vector $h^{NMF-NMF}(I)$ by applying NMF–NMF algorithm;

Step 2 : Pseudorandom weight vectors $\{t_i\}_{i=1}^{M}$ are generated, where M is the length of the final hash. The final hash vector is given by $\{< h^{NMF-NMF}(I), t_1 >, ..., < h^{NMF-NMF}(I), t_M >\}$, where $< x, y >$ denotes the inner product of vectors x and y;

In the verification stage, the comparison of the hash vectors obtained for the original and the candidate image is done based on the L_2 norm between the two hashes.

The NMF hashing method possesses excellent robustness under a large class of perceptually insignificant attacks like JPEG compression, shearing, γ correction, cropping, rotation, affine wrapping while reducing misclassification rate for perceptually different images.

Inspired by the promising results obtained using dimension reduction techniques, Lv et colleagues propose in [Lv 2009] a new robust and secure image hashing scheme based on Fast Johnson–Lindenstrauss Transform (FJLT). By using the FJLT, n points from the Euclidean space can be projected from the original d dimension space down to k dimension space, with an error of ε. The FJLT, denoted $\Phi = FJLT(n, d, \varepsilon)$, is given by:

$$\Phi = P \cdot H \cdot D \tag{2.15}$$

where P is a matrix of size k x d with components distributed according to a normal distribution with zero–mean and variance q^{-1}, H is a normalized Hadamard matrix of size d x d, and D is a d x d diagonal matrix whose elements are distributed uniformly in the interval [-1,1].

The authors chose FJLT to reduce dimension because, being a random projection, it enhances the security of the hashing method and also because FJLT is robust to most routine degradations and malicious attacks.

The proposed hashing scheme based on FJLT is divided into three steps: random sampling, dimension reduction by FJLT, and ordered random weighting as described below:

Step 1 *Random Sampling* consists in selecting a limited number of subimages as original features. The novelty of the Lv *et al.* approach is that subimages are considered as points in high–dimensional space. For example, the sub–images in Fig.2.10 are squares of size p x p which can be represented as points in a p^2–dimensional space.

The original feature is constructed by concatenating the columns of the N subimages R_i, $i = 1, ..., N$, which were pseudo–randomly selected from the gray–level input image by using a secret key:

$$Feature = \{R_1, R_2, ..., R_N\} \tag{2.16}$$

The advantage of this approach is that the Feature matrix contains global information of the image, while the the components R_i contain local information, being thus resistant to geometric attacks like cropping.

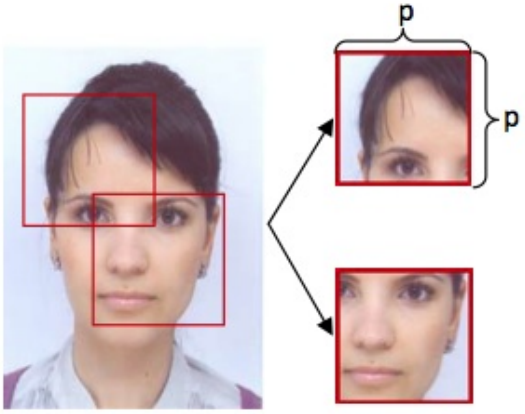


Figure 2.10: Random sampling.

Step 2 *Dimension Reduction*: by using FJLT, the essential features of the original data are captured in a lower–dimensional space. In this step, the authors map the above Feature matrix from a p^2 x N space to a k x N space. The intermediate hash IH is defined as

$$IH = \Phi(Feature) = P \cdot H \cdot D \cdot Feature \tag{2.17}$$

where P, H and D are the matrices used in 2.15. The dimension k can be obtained by tuning the number of subimages.

Step 3 *Ordered Random Weighting* is necessary because, even after dimension reduction, the intermediate hash can be too large. The hash is computed as:

$$Hash = \{< IH_1, w_1 >, ..., < IH_N, w_N >\} \tag{2.18}$$

where IH_i is the i^{th} column of IH, $\{w\}_{i=1}^{N}$ are the pseudorandom weight vectors, and $< a, b >$ is the inner product of vectors a and b.

The hash obtained by using this scheme is an intermediate hash, but a final hash value can be obtained by using a compression step as in [Swaminathan 2002, Monga 2006].

Because the FJLT hashing scheme is sensitive to rotations, the authors propose a rotation invariant Fast Johnson–Lindenstrauss Transform (RI–FJLT) hashing scheme . The steps of the proposed approach can be summarized as follows:

1) Convert an input image into log–polar coordinates:

$$I(x, y) \rightarrow J(log\rho, \theta) \tag{2.19}$$

where x and y are Cartesian coordinates and ρ and θ are Log–Polar coordinates. By changing the representation system, rotation and scaling will be horizontal and vertical offsets as it can be seen in Fig.2.11.

2) Apply Mellin Transform (MT) to image J in order to obtain the magnitude feature image;

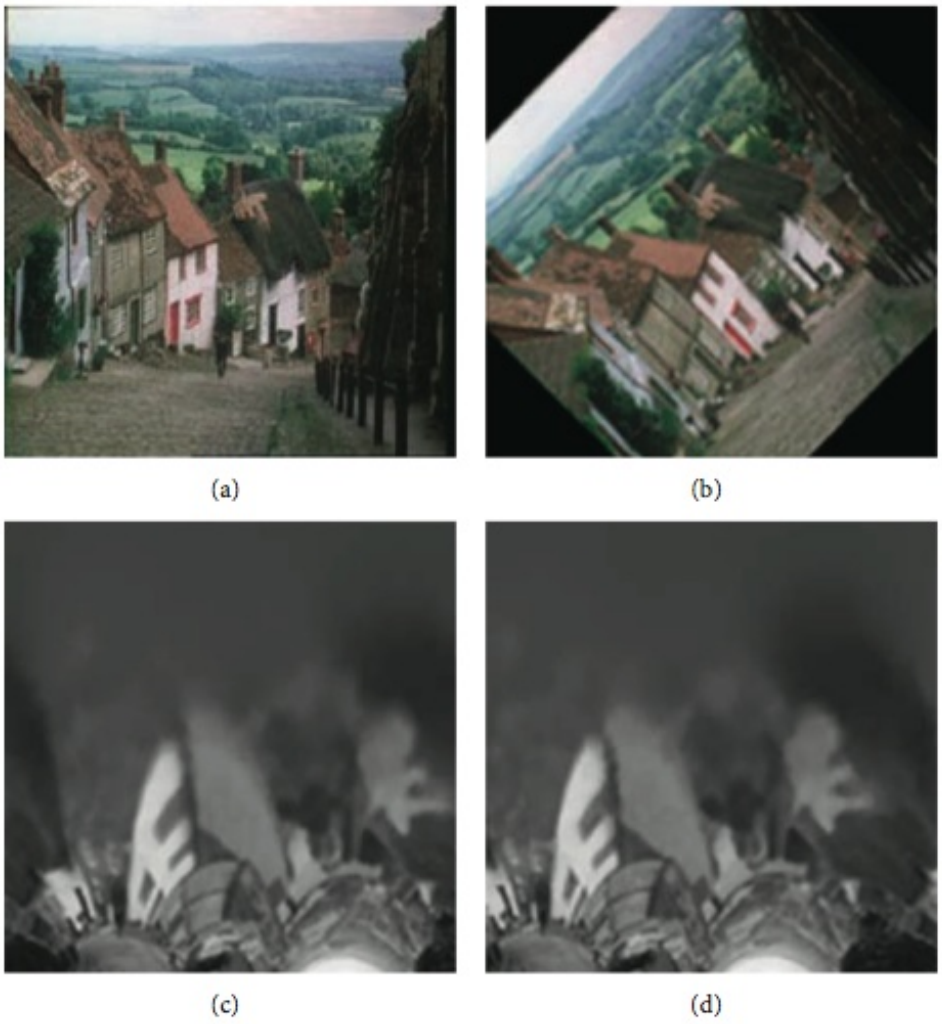


Figure 2.11: Example of how rotation attack affects an image in Cartesian and Log–polar system: a) Original image in Cartesian system; b) Rotated image with 45 degrees in Cartesian system; c) Original image in Log–polar coordinates; d) Rotated image in log–polar coordinates.

3) Obtain the hash value by applying the FJLT hashing scheme described above on the magnitude feature vector;

In the identification stage, for a candidate image I_c, the hash value is computed. Then the Euclidean distances to each original image existing in the database are calculated. The candidate image I_c is identified as being the one that yields the minimum corresponding distance.

The FJLT hashing scheme achieves similar or better performance than the NMF hashing scheme under manipulations and attacks such as additive noise, blurring, geometric attacks, JPEG compression and Gamma correction. Poor performance is achieved only in the case of the rotation attacks. This deficiency is remedied by incorporating MT into the algorithm.

2.3.5.2 Image hashing in print–and–scan scenario

The print–and–scan attack is a difficult problem not yet solved in image hashing. In the literature, few image hashing algorithms have been tested against print–and–scan attacks.

In 2007, Yu *et al.* have proposed in [Yu 2007] an image hashing algorithm robust to print–and–scan attack. The method is based on the algorithm proposed in [Monga 2005]. In the hash extraction step, the authors replace the end–stop wavelet, employed in [Monga 2005], with the extractor proposed in [Zitova 1999].

In the process of structure matching a modified version of the Hausdorff distance, proposed by Monga *et al.* in [Monga 2005] is used for comparing the hashes of two images. If the distance is smaller than a predefined threshold ε, then the image is considered authentic.

The algorithm proved to be robust to print–and–scan attack, compression, common signal processing operations, global and local geometric transformations.

In [Wu 2009], Wu *et al.* propose an image hashing algorithm robust to print–and–scan and sensitive to content changing.

Wu *et al.* analyzed the print–and–scan attack and divided them into three types:

- *degrading attacks* such as filtering and noise adding, which modify pixels independently;
- *global attacks* such as luminance adjustment, which modify pixels uniformly;
- *geometric attacks* such as rotation and scaling, which modify pixels position;

In order to face these three types, they suggest to extract the feature as the relationship of the Radon coefficients after applying WT. The feature extraction is presented in Fig. 2.12.

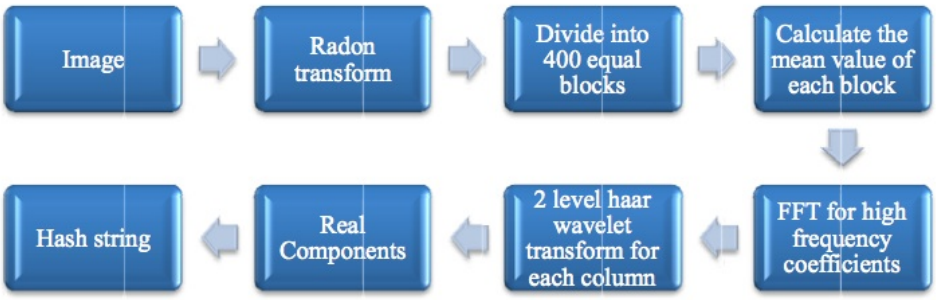


Figure 2.12: Wu *et al.* proposed scheme for image hashing.

Firstly, the Radon transform is applied on image I. Before feature extraction, some pre–processing steps are needed:

- deleting the rows containing "0"'s in order to remove the redundancy introduced by rotation and shifting;

- locating two points A and B in the Radon matrix and cyclically shift the matrix vertically until the two points are situated in the middle row;

Next, the Radon transform is divided in 40 x 20 blocks and the mean of each block is computed. A 2–level wavelet decomposition is employed to obtain the features from the Radon coefficients. On the high frequency wavelet coefficients the Fast Fourier Transform (FFT) is performed and only the real components are retained to construct the hash vector. The final hash is obtained by binarizing the FFT coefficients by using as threshold the coefficients mean.

This algorithm presents advantages such as security, good discrimination and especially robustness to the print–and–scan attack.

2.4 Conclusion

In this chapter several types of image hashing algorithms have been presented. The two main domain applications, indexing and authentication, have also been defined. It has been shown how hash functions provide an efficient way of protecting image content–integrity and which are the desirable properties of a hash in order to attain robustness and security.

Various algorithms for constructing image hashes schemes able to survive to non–malicious degradations and malicious attacks have been proposed in the literature, but there is no universal hash robust to all types of attacks. Some of the most important problems when designing a hash scheme include the extraction of perceptually more robust features, suitable distance metric, used to measure the similarity between the hashes of two images, more sophisticated strategies for introducing secret keys, etc.

In this chapter an attempt to classify the large number of existing algorithms has been made. The first perceptual image hashing schemes, based on histogram intensity, proved to be insecure. Several hash algorithms based on DCT/DWT, low–level feature extraction, relation–based, proved to be resistant to attacks such as JPEG compression, blurring, scaling, etc. But the recently developed algorithms, based on dimension reduction techniques, outperform the former algorithms regarding robustness. Even if these algorithms report good results for JPEG compression or rotation, few of them have been tested in print–and–scan scenario.

Chapter 3

Independent Component Analysis

3.1 Introduction

This chapter is an introduction to the concept of ICA. ICA is a variant of PCA in which the components are supposed to be statistically independent. Section 3.2 presents PCA as a powerful tool for analyzing images. It consists in feature extraction and image compression by reducing the number of dimensions. The main steps needed when performing PCA are also described in this section.

Foundations and basic knowledge necessary to understand ICA are presented in section 3.3. Several algorithms for achieving statistically independent components are described in section 3.4. Section 3.5 summarizes the main ideas of this chapter.

3.2 Principal Component Analysis

PCA represents a classic statistical technique for feature extraction and compression which was introduced by Pearson in 1901 [Pearson 1901]. The goal is to reduce the redundancy from large data sets by using the correlation between data elements as a measure. Thus, the PCA algorithm is based only on the first and the second order statistics.

Let us consider a data set formed by a random vector X.While the first and second order statistics are known or can be estimated, no explicit assumption on the probability density is made in PCA. First, the variables in X are centered by subtracting the mean and the variance is set to 1 by normalization:

$$\begin{aligned} centering\ X &\leftarrow X - E\{X\} \\ normalization\ X &\leftarrow X/std(X) \end{aligned} \tag{3.1}$$

Next the vector X is linearly transformed into a vector P so that the redundancy due to correlation is removed. The transformation consists of finding a rotated orthogonal coordinate system so that the elements of X in the new coordinate system should become uncorrelated [Hyvärinen 2001]. The first axis corresponds to the maximal variance of the projection of X on the new coordinate axes, the second axis to the maximal variance

in the direction orthogonal to the first axis. The components which have very small variance values can be discarded, while the rest form the elements of vector P. Thus, the first component extracted has the maximal amount of the total variance in the observed variables, where the total variance accounts for the sum of the variance of each element in X.

The PCA algorithm can be summarized by the following steps:

a) Centering and normalization of the data set as described in 3.1;

b) Computation of the covariance matrix:

$$Cov_X = E\{XX^T\} \tag{3.2}$$

c) Calculation of the eigenvalues $d_1, d_2, ...d_n$ and eigenvectors, $C_1, C_2, C_3, ..., C_n$ of the covariance matrix;

d) Choosing the components and forming the P vector. The eigenvectors, $(C_1, C_2, C_3, ..., C_n)$, obtained at the previous step are ordered as columns of a matrix from the highest to the lowest eigenvalue, $d_1, d_2, ...d_n$. Thus, the first principal component of X is given by :

$$P_1 = C_1^T X \tag{3.3}$$

The eigenvalues of Cov_X denotes the variance of the corresponding principal component and can be easily proven by:

$$E\{P_m^2\} = E\{(C_m^T X)^2\} = E\{C_m^T X X^T C_m\} = C_m^T E\{XX^T\} C_m = C_m^T Cov_X C_m \tag{3.4}$$

By taking into account that the mean of the original data set, X is zero, E{X}=0, then the principal components have also zero mean, i.e. $E\{P_m\} = C_m^T E\{X\} = 0$. Thus, a small value of the variance (a small eigenvalue, d_m) indicates that the principal component P_m is close to zero.

An important practical problem is how to determine the number of principal components which should be retained. The error between the new compressed data set obtained after PCA algorithm was applied, $\hat{X}$, and the original data set, X can be expressed as following:

$$E\{||X - \hat{X}||^2\} = \sum_{i=m+1}^{n} d_i \tag{3.5}$$

where m is the number of principal components selected. Hence, the error is the sum of the eigenvalues corresponding to the discarded eigenvectors, $C_{(m+1)}, C_{(m+2)}, ..., C_n$. If $m = n$ the error is zero; it increases as more and more principal components are discarded. In other words a trade–off has to be made between the error and the amount of data needed for expansion. One of the most straightforward method for deciding the number of the principal components to be retained is the variance criterion. The eigenvalues of the covariance matrix are usually sharply decreasing, thus, a lower limit for the eigenvalue can be set so that the verry small principal

components are discarded. The computation of the threshold can be determined by using a priori information about X, if it is available. However, a good value for m is obtained by using the Akaike's information criterion and the minimum description length criterion.

e) *Deriving the new data set*: The original data set X can be expressed by using the PCA truncated form:

$$\hat{X} = \sum_{i=1}^{m} P_i C_i \tag{3.6}$$

There are different algorithms described in the literature for finding the computation of the PCA, like: neural networks [Haykin 1998], recursive least–squares approach (the PAST algorithm) [Haykin 1998], etc.

3.3 Independent Component Analysis

The linear statistical model for ICA is a factorisation:

$$X = AS \tag{3.7}$$

The objective of ICA is the estimation of the mixing matrix A and of the independent components S based on the observation vector X and under the assumptions that [Hyvärinen 2001]:

- The columns of the mixing matrix A are linearly independent;
- The elements of the vector S, i.e. the independent components, are mutually independent and also independent from the components;
- The independent components must have a nongaussian distribution; it can be easily proven that, in the case of independent components with Gaussian distribution, the orthogonal mixing matrix cannot be estimated because the distribution of the observations is the same as the one for the independent components, regardless the orthogonal mixing matrix A. Moreover, ICA is based on higher order cumulants which for a Gaussian distribution are zero, thus ICA is essentially impossible if the sources have Gaussian distributions;

Conceptually, ICA is "stronger" than PCA because, while the latter is based only on second order cumulants, ICA retains higher order cumulants, especially the kurtosis (4th cumulant). Basically, ICA succeeds where PCA fails because statistical independence is "stronger" than uncorrelatedness and whitening. Hence, while two random independent variables are also uncorrelated, contrariwise uncorrelatedness does not imply independence. Moreover, whiteness of a random vector h with zero mean represents the fact that the elements of the vector are uncorrelated and have unit variance, i.e. the covariance matrix is equal with unit matrix: $E\{hh^T\} = I$.

Whitening can be obtained by using the SVD or Eigenvalue Decomposition (EVD). Thus, whitening can be expressed as follows:

$$\tilde{X} = TAS = \tilde{A}S \tag{3.8}$$

where $\tilde{A}$ is the new mixing matrix which whitens the observation vector X and T is a linear transformation applied in order to obtain the whitened version of X, $\tilde{X}$.

However, as stated before, the whitening is not sufficient for estimating the ICA model, because e.g. if we consider an orthogonal transformation Y for $E\{\tilde{X}\}$ then:

$$y = YE\{\tilde{X}\} \tag{3.9}$$

$$E\{yy^T\} = \{Y\tilde{X}Y^T\tilde{X}^T\} = YE\{\tilde{X}\tilde{X}^T\}Y^T = YIY^T = I \tag{3.10}$$

A consequence of equation 3.10 is that the vector y is also white, thus it cannot be distinguished whether the independent components are given by $E\{\tilde{x}\}$ or y; because y can be any orthogonal transformation of $\tilde{X}$, whitening is able to offer the independent components up to an orthogonal transformation [Hyvärinen 2001].

However, whitening can be used to reduce the complexity of the ICA problem; thus, it is very useful as a preprocessing step for ICA, because it gives an orthogonal matrix, $\tilde{A}$ for A. Hence, the complexity of the problem reduces from having to estimate m^2 parameters (in the case $m = p$) to the case where only an orthogonal mixing matrix $\tilde{A}$ which contains $m(m-1)/2$ degrees of freedom has to be estimated.

The estimation of the independent components is reduced to:

$$S = WX, \qquad W = A^{-1} \tag{3.11}$$

There are different approaches for the estimation of all the parameters of the ICA model:

- *Maximization of nongaussianity*: according to the central limit theorem, the sum of two independent variables is more Gaussian than the original variables. Thus, the sum of the independent components, S, is more gaussian than every s_i and it becomes the least gaussian when it equals one of the independent components s_i (this is true if the s_i have the same distribution [Hyvärinen 2001]); In order to measure the nongaussianity, the fourth order cumulant (kurtosis) is introduced:

  $$kurt(X) = E\{X^4\} - 3E\{X^2\}^2,\ E\{X^2\} = 1 \Rightarrow kurt(X) = E\{X^4\} - 3 \tag{3.12}$$

  where X is centered and normalized. The absolute value or the square value of the kurtosis is used for measuring the nongaussianity. Thus, Gaussian random variables have a kurtosis equal to zero while for most of the nongaussian random variables the kurtosis is greater than zero. An algorithm which uses the kurtosis to estimate the independent components is the Fast Independent Component Analysis (FastICA) algorithm introduced by [Hyvärinen 1997]. The advantage of this algorithm is the very fast and reliable convergence in comparison with gradient–based algorithms.

  However, the kurtosis is very sensitive to outliers making it not a robust measure of nongaussianity. Another measure of nongaussianity which is opposite to kurtosis

in many ways, i.e. it is robust but complicated, is the negentropy. It is based on the differential entropy from information theory and on the fact that a gaussian random variable has the largest entropy among all variables with equal variance. The negentropy can be defined as follows:

$$J(X) = H(X_{gauss}) - H(X) \tag{3.13}$$

where X_{gauss} is a gaussian random variable with the same covariance matrix of the random variable X and $H(X)$ is the entropy. $J(X)$ is nonnegative and it is equal with zero if and only if X has a gaussian distribution. However, because the negentropy is complicated to determine an estimate of higher order cumulants is usually used:

$$J(X) \approx \frac{1}{12} E\{X^3\}^2 + \frac{1}{48} kurt(X)^2 \tag{3.14}$$

- *Maximum likelihood estimation*: Pham et al. [Pham 1992] formulated the likelihood in the noise free ICA model according to:

$$L = \sum_{k=1}^{K} \sum_{i=1}^{m} \log f_i(w_i^T X(k)) + K \log |detW| \tag{3.15}$$

where $W = (w_1, w_2, ..., w_m)^T$ is the inverse matrix of A, f_i are the density functions of the independent components s_i, K are the samples for observations $X(k)$;

There are different algorithms which use the likelihood maximization for solving the ICA model:

  - The Infomax algorithm [Bell 1995] - based on maximizing the output entropy of a neural network with non–linear outputs. It was demonstrated that the maximum likelihood estimation is equivalent with the infomax principle, i.e. maximization of the network entropy [Cardoso 1997].
  - A fast fixed–point algorithm: the FastICA algorithm introduced above for nongaussianity maximization can be directly applied to the maximization of the likelihood.

- *Minimization of mutual information*: Mutual information of a random vector, x, with random components variables $x_i, i = (1..m)$ is:

$$I(X_1, X_2...X_m) = \sum_{i=1}^{m} H(X_i) - H(X) \tag{3.16}$$

It measures the mutual dependence between two variables and it can also be interpreted as the code length. Hence, $H(X_i)$ represents the code length for each element in vector X, while $H(X)$ is the code length of the random vector X. Mutual information is a measure that shows an increase in code length if the coding is done on each component of X compared with the case of coding the global vector X. However, in the case where the X_i are independent, coding each element separately

would not increase the code length. Thus, in order to solve the ICA model, the matrix W from equation 3.11 is estimated in such a way that the mutual information of s_i is minimized. It can be demonstrated that the estimation of matrix W is roughly equivalent to the maximization of the negentropy. The mutual information is:

$$I(S_1, S_2...S_m) = \sum_{i=1}^{m} H(S_i) - H(X) - log|detW| \tag{3.17}$$

If the components of vector X are uncorrelated and have unit variance, then: $E\{SS^T\} = WE\{XX^T\}\ W^T = 1$, thus:

$$I(S_1, S_2...S_m) = const. - \sum_{i=1}^{m} (S_i) \tag{3.18}$$

Equation 3.18 shows the fundamental relation between mutual information and negentropy.

Moreover, the maximum likelihood and the mutual information are two interconnected concepts , hence, it can be proved that the mutual information can be expressed [Hyvärinen 2001] through the following formula:

$$\frac{1}{K} E\{L\} = \sum_{k=1}^{K} E\{\log f_i(w_i^T X) H(X_i) - H(X) \tag{3.19}$$

- *Tensorial methods*: These types of methods are based on the fact that the eigenvalues of the cumulant tensors give the independent components. The fourth–order cumulant is a four–dimensional array whose entries are given by the fourth–order–cross–cumulants of the data [Hyvärinen 2001]. It can be considered as a forth–order matrix. It also can be proven that the eigenvalues of the cumulant tensor given by the kurtosis of the independent components are non–zero while the rest are zero. The Joint Approximate Diagonalization Eigenmatrices (JADE) is a method which shows how the cumulant tensors can be computed and which also deals with the special case where the eigenvalues of the cumulant tensor are not distinct. A more detailed description of the algorithm can be found in [Cardoso 1997]. Other methods based on the cumulant tensors are High Order Singular Value Decomposition (HOSVD) [de Lathauwer 1995] and Higher Order Eigenvalue Decomposition (HOEVD) [Comon 1994].

- *Other Methods*: for solving the ICA model are:

  - *ICA based on nonlinear decorrelation and nonlinear PCA* [Herault 1984]. It is not used anymore in practice because it is not as efficient in comparison with the new algorithm presented above;
  - *Methods that use time structure*: in this case the ICA model is a little bit different from the basic model because the samples of vectors X and S are not random but have a specific order. Thus, these methods are based on time

dependencies of the independent components using the autocovariance and variance non stationarities as measures [Hyvärinen 2001]. A simple algorithm using the time dependencies through autocovariance is AMUSE [Tong 1991]. However, this simple and fast algorithm is not able to discriminate between independent components associated with identical eigenvalues of the covariance matrix of the whitened data set. Another algorithm which takes into consideration the variance of non stationarities in order to solve the ICA model can be derived from the FastICA [b) [Hyvärinen 2001]. Other algorithms using the periodicity of the signals are proposed in [Tsalaile 2009]. Nevertheless, the algorithms described in the basic ICA model can be used also, but because they do not use the whole data structure they may converge far from the optimal solution.

3.4 Various ICA algorithms

3.4.1 FastICA

In practice, algorithms for maximizing/minimizing the contrast function are needed. A very efficient method of non–gaussianity maximization based on a fixed–point iteration is FastICA [Hyvärinen 1997].

3.4.1.1 FastICA for one unit

FastICA for one unit refers to maximizing the non–gaussianity of $w^T X$ in order to obtain one independent component. The weighted vector w represents one of the rows of matrix A. The non–gaussianity is measured by the approximation of negentropy J($w^T X$) given by :

$$J(w^T X) \approx [E\{G(w^T X)\} - E\{G(v)\}]^2 \tag{3.20}$$

where v is a Gaussian variable of zero mean and unit variance.

The FastICA can be derived as an approximative Newton iteration. Let g be the derivative of the non–quadratic function G. The main steps of the FastICA algorithm can be summarized as follows:

Step 1 Choose an initial random weight vector w.

Step 2 Let $w^+ =$ E{Xg($w^T X$)}-E{$g'(w^T X)$}w.

Step 3 Let $w = w^+/||w^+||$.

Step 4 If the convergence is not reached, go to *Step 2*.

3.4.1.2 FastICA for several units

The one–unit algorithm presented above estimates only one of the independent components. In order to estimate more independent components, the FastICA algorithm for one–unit must be applied several times for different weight vectors $w_1, ..., w_n$.

In order to prevent that different weighted vectors $w_1, ..., w_n$ converge to the same maxima, a decorrelation step of the outputs $w_1^T X, ..., w_n^T X$ after every iteration must be introduced. Two methods for achieving de–correlation are presented below.

The first method is a deflation scheme based on a Gram–Schmidt–like decorrelation. In this approach the independent components are estimated one by one. When p independent components, or p vectors $w_1, ..., w_p$, have been estimated, the one–unit fixed–point algorithm is performed for w_{p+1}. After every iteration step, the projections $w_{p+1}^T w_j w_j, j = 1, ..., p$ are extracted from w_{p+1} and then w_{p+1} is renormalized:

Step 1 Let $w_{p+1} = w_{p+1} - \sum_{j=1}^{p} w_{p+1}^T w_j w_j$

Step 2 Let $w_{p+1} = w_{p+1} / \sqrt{w_{p+1}^T w_{p+1}}$

A second method is a symmetric decorrelation–based method, in which no vectors are "privileged" over others [Karhunen 1997]. This can be accomplished by using the classical method involving matrix square roots:

$$W = (WW^T)^{-1/2} W$$

where W is the matrix of vectors $(w_1, ..., w_n)^T$ and the inverse square root $(WW^T)^{-1/2}$ is obtained from the eigenvalue decomposition of WW^T.

A simpler alternative is the iterative algorithm proposed in [Hyvärinen 1999]:

Step 1 Let $W = W / \sqrt{||WW^T||}$

Step 2 Let $W = \frac{3}{2} W - \frac{1}{2} WW^T W$

Step 2 is repeated until convergence is achieved. The norm in *Step 1* is an ordinary matrix norm, i.e. the 2–norm or the largest absolute row (or column) sum.

When compared with the existing methods for ICA, the FastICA algorithm presents certain desirable properties as summarized below:

1. Cubic or at least quadratic convergence versus linear convergence for ordinary ICA;
2. The algorithm is easy to use; unlike gradient–based algorithms, no step size parameters have to be chosen;
3. The algorithm extracts directly independent components of any non-Gaussian distribution using any nonlinearity g;
4. The performance of the method can be increased by choosing a suitable nonlinearity g;
5. Its computational load is decreased because the independent components can be estimated one by one;
6. The FastICA algorithm is parallel, distributed, computationally simple and needs little memory space.

3.4.2 EFICA

Efficient Fast Independent Component Analysis (EFICA) is an improved version of the FastICA algorithm, which is asymptotically efficient, i.e. its accuracy attains the Cramér-Rao lower bound. The proposed EFICA algorithm in [Koldovsky 2006] has a computational complexity three times higher than that of ordinary symmetric FastICA. The extracted independent components are modeled as Generalized Gaussian Distribution (GGD) with adequate parameters.

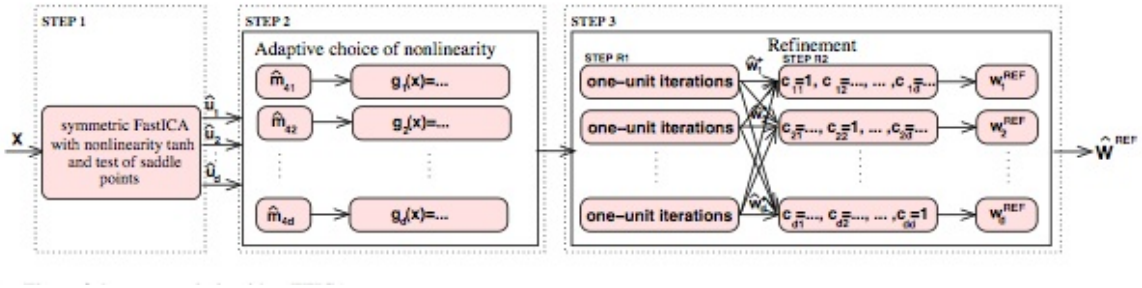


Figure 3.1: EFICA algorithm proposed in [Koldovsky 2006].

3.4.2.1 Algorithm

The proposed algorithm associates the idea of the generalized symmetric FastICA with an adaptive choice of the nonlinearity function g [Koldovsky 2006]. The algorithm contains three steps:

Step 1 consists in performing the original symmetric FastICA with a standard function g until convergence is reached;

Step 2 consists in an adaptive choice of the nonlinearity g_k in order to estimate the cost function of the extracted components;

Step 3 consists in a refinement of each of the independent components extracted by using a general version one–unit FastICA algorithm proposed in [Koldovsky 2006]. Instead of one nonlinearity g, the general version uses the different nonlinearities g_k obtained in *Step 2*;

3.4.3 InfoMax

InfoMax is a method which maximizes the entropy of the auxiliary outputs y. y is defined as:

$$y = g(S) \tag{3.21}$$

The maximization of the joint entropy of two outputs consists of maximizing the individual entropies while minimizing the mutual information [Bell 1995]. When the mutual information is zero, the two outputs are statistically independent.

3.4.3.1 Algorithm

The InfoMax is a gradient–based algorithm, which maximizes entropy. The gradient of the unmixing matrix W is given by:

$$\nabla W = [W^T(k)]^{-1} + [I - 2y(k)]X^T(k) \tag{3.22}$$

3.4.4 Pearson ICA

Pearson ICA [Karvanen 2000] is a BSS method combining the Pearson system, based on the maximum likelihood approach, with first contrast functions.

The Pearson system is employed in order to model the distributions of the independent components. The Pearson system is defined by:

$$f'(x) = \frac{(x-a)f(x)}{b_0 + b_1 x + b_2 x^2}, \tag{3.23}$$

where a_0, b_0, b_1 and b_2 are the parameters of the distribution. The cost function of the Pearson system is obtained from 3.23:

$$\varphi(x) = -\frac{f'(x)}{f(x)} = -\frac{x-a}{b_0 + b_1 x + b_2 x^2}. \tag{3.24}$$

The parameters a_0, b_0, b_1 and b_2 can obtained from the method of moments. Parameters a_0, b_0, b_1 and b_2 can be expressed as functions of second μ_2, third μ_3 and fourth μ_4 central moments of the distribution:

$$b_1 = a = -\frac{\mu_3(\mu_4 + 3\mu_2^2)}{C} \tag{3.25}$$

$$b_0 = -\frac{\mu_2(4\mu_2\mu_4 - 3\mu_3^2)}{C} \tag{3.26}$$

$$b_2 = -\frac{(2\mu_2\mu_4 - 3\mu_3^2 - 6\mu_2^3)}{C}, \tag{3.27}$$

where $C = 10\mu_4\mu_2 - 12\mu_3^2 - 18\mu_2^3$.

3.4.4.1 Algorithm

Karvanen and colleagues consider as contrast function the cost functions. Since the independent components distributions are known, the cost functions are the optimal choice for the contrast function.

In the Pearson ICA algorithm, the independent components distributions are estimated by fitting the marginal distributions into the Pearson family by using the method of moments.

The steps of the Pearson ICA algorithm can be summarized as follows:

Step 1 : Select as contrast function either the contrast function of the Pearson system or the hyperbolic tangent. The contrast function selection is dependent of the third and fourth moment;

Step 2 : If the Pearson system was selected, the parameters of the distribution are estimated by using the method of moments;

Step 3 : Compute the cost function ϕ for the Pearson system or for the hyperbolic tangent;

Step 4 : Compute the demixing matrix W;

The four steps are repeated until convergence is achieved.

3.5 Conclusion

This chapter presents the concept of ICA as a fine–tuning of PCA. Both PCA and ICA are powerful tools for feature extraction and for dimension reduction of an observed number of variables.

PCA is an unsupervised statistical method that includes a preliminary learning stage. In this stage PCA derives a basis of uncorrelated eigenvectors. The number of eigenvectors may be reduced by discarding the ones corresponding to the least eigenvalues.

A drawback of PCA is that it only takes the second–order statistics into account. ICA was developed to exploit higher–order statistics. ICA recovers statistically independent components from a linear mixture. In order to attain independence, several algorithms that optimize a specific cost function were implemented. Many of them use PCA, called also weak independence. They iteratively optimize the cost function, whose global optimum occurs when independence is achieved. For instance, InfoMax maximizes the entropy of the estimated independent components, FastICA maximizes their non–Gaussianity and Pearson–ICA algorithm minimizes the mutual information of the sources.

Chapter 4

ICA Arhitecture I vs. ICA Arhitecture II

4.1 Introduction

In this chapter, two different ways of applying ICA on ID images are described. The two representations proposed by Bartlett in [Bartlett 2002] are described in Section 4.2. Depending on how ICA is applied on the data set, either statistically independent basis vectors or statistically independent projection coefficients are obtained. In both cases, a basis of images for projection is obtained. The proposed method for the verification of ID images based on the two architectures is described in Section 4.3 and different algorithms for selecting the most relevant components for the subspace projection are proposed in Section 4.4. In Section 4.5, the performance of the two representations on simulated data are compared. Section 4.6 concludes the chapter by summarizing the central ideas.

4.2 ICA Architectures

4.2.1 Generalities

The two different ways of applying ICA to ID images are known in the literature ([Bartlett 2001], [Bartlet
as ICA architecture I and ICA architecture II.

4.2.1.1 Architecture I

In architecture I, ICA considers different images as random variables and pixels as variable samples. The process is described in Fig. 4.1. The input face images, serialized and placed on the different rows of matrix X, are considered to be a linear combination of independent components S. The ICA algorithm approximates the weight matrix W in order to recover the statistically independent components placed onto the rows of U, the matrix of the estimated components. These components represent the basis images employed in representing ID images. The first row of Fig. 4.2 shows a subset of eight basis images obtained with architecture I. They are localized feature vectors (i.e. face

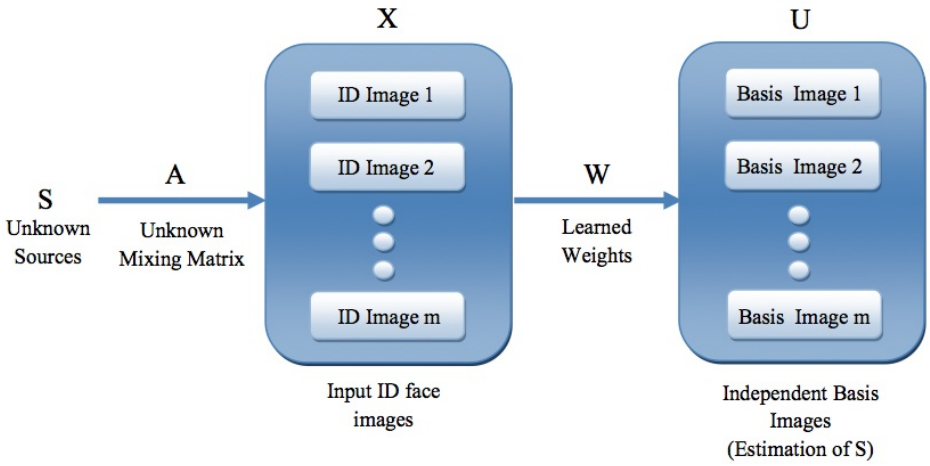


Figure 4.1: ICA architecture I: finding statistically independent basis images.

characteristics) influenced only by small areas of the image. The ID image representations are vectors of coordinates obtained by projecting these images with respect to the basis defined by the rows of matrix U as in Fig. 4.3. These vectors are situated in the rows of mixing matrix $A = W^{-1}$.

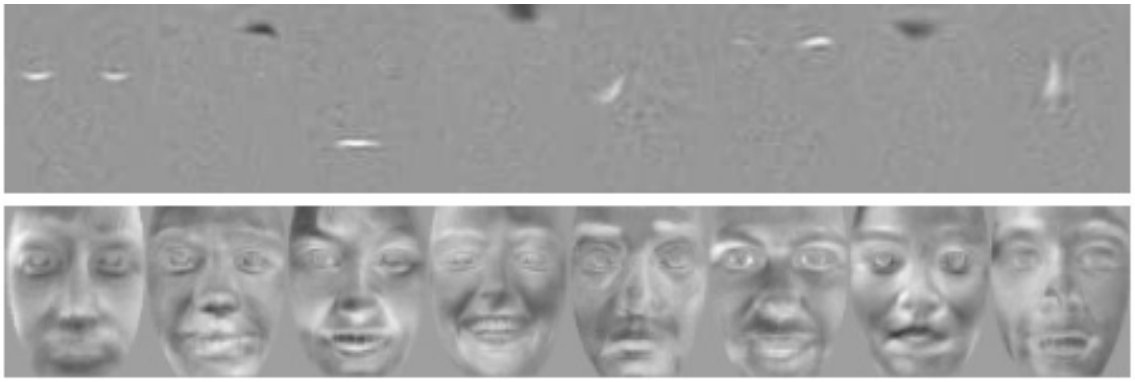


Figure 4.2: Basis vectors. On the first row, respectively on the second, an example of basis vector for Architecture I, respectively for Architecture II.

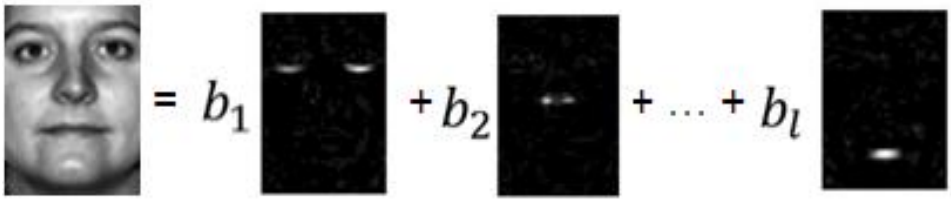


Figure 4.3: Face representation for ICA architecture I.

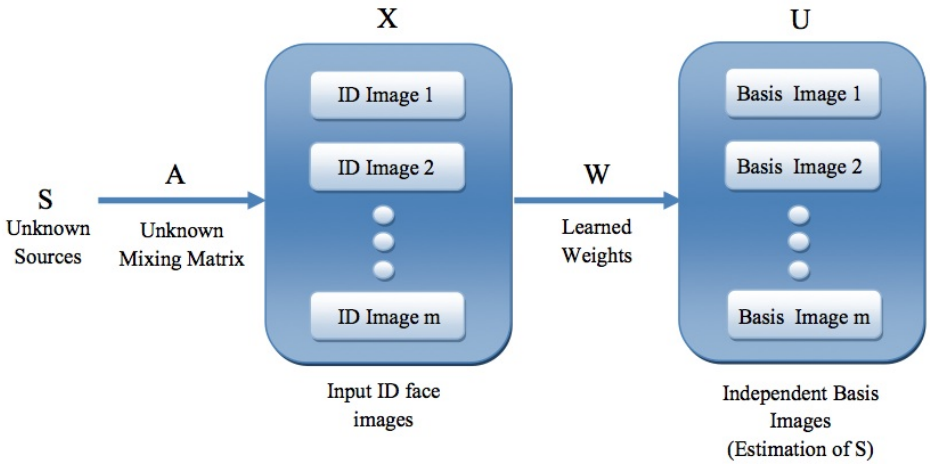


Figure 4.4: ICA architecture I: finding statistically independent coefficients.

4.2.1.2 Architecture II

Even if the basis images are statistically independent, the coordinates that represent the ID images in the subspace defined by ICA architecture I are not. The purpose of ICA architecture II is to obtain statistically independent coordinates employed to represent the input ID images. For this purpose, the ID images are serialized and organized onto the columns of matrix X; in this way pixels are considered as random variables and different images as variable samples. Thus the columns of A, the pseudo–inverse of the weight matrix W, are considered as basis images. The independent coefficients are recovered in the columns of U(Fig.4.4). They represent the coordinates employed to reconstruct the input ID images (Fig. 4.5).

Figure 4.5: Face representation for ICA architecture II.

Eight basis images obtained with architecture II are shown in the second row of Fig. 4.2. They represent global features vectors unlike the ones obtained when using architecture I.

4.2.2 Mathematical Background

Bartlett and colleagues propose in [Bartlett 1998], [Bartlett 2002] to perform ICA as a two–step process in order to have control on the number of independent components

allowing thus to create projection subspaces of size l, for any $l \in \mathbb{N}$. This is done by using PCA as a pre–processor and then by applying ICA on the resulted eigenvectors instead of the input ID images. The replacement of the original ID images with the first l PC eigenvectors will not affect the process. Even more, it increases ICA performances by reducing dimensionality and computational complexity as argued in [Liu 1999].

4.2.2.1 Architecture I

In this section, the mathematical basis of the process depicted in Fig. 4.1 are described. Let C be a m x l matrix containing the first l Principal Component (PC) vectors of n input ID images and m the number of pixels in an image. Because in this representation the rows of matrix X are variables and the columns are samples, ICA is applied on C^T. The l basis images from U used to represent the input ID images are determined as follows:

$$P = X * C \rightarrow X = P * C^T \tag{4.1}$$

where P is the matrix of PCA coefficients.

We know that ICA is applied on the transposed of C:

$$U = W * C^T \tag{4.2}$$

From 4.1 and 4.2 we get

$$X = (P * W^{-1}) * U = B * U \tag{4.3}$$

Therefore the coordinates used to represent the input ID images are given by the rows of the matrix

$$B = P * W^{-1} \tag{4.4}$$

The coordinates for the test ID images were determined by using a PC representation based on the training image set as defined in:

$$B_{test} = P_{test} * W^{-1} \tag{4.5}$$

Without PCA pre–processing, $B = W^{-1}$ and $B_{test} = X_{test} * U^T$.

4.2.2.2 Architecture II

In architecture II, ICA tries to find independent coefficients for the input images placed on the columns of matrix U. Also in this approach, ICA is performed on the PCA coefficients instead of the original ID images, in order to reduce dimensionality [Bartlett 1998], [Bartlett 2002]. Using the same notations as before:

$$U = W * P^T \tag{4.6}$$

The ICA representation for the test ID images are placed in the columns of U as follows:

$$U_{test} = W * P_{test}^T \tag{4.7}$$

The basis vectors are obtained in the columns of $A = W^{-1}$. Without PCA pre–processing, $U_{test} = W * X_{test}^T$

4.3 Our method for ID Image Verification

In this section I explain the proposed method for ID image verification applications. The goal is to obtain an optimal ID image representation by using ICA in order to be robust to print–and–scan attacks (a primary non–malicious attack that the ID image verification system must take into account).

The application is a two step–process consisting in the following stages: learning, enrollment and verification. In the first stage the image basis used for projecting the images is computed. The enrollment stage, shown in Fig. 4.6, is a three–step process: capture, process and enroll. The ID image is captured by using a sensing device (scanner, camera, etc.). Further, the face is cropped from the image, processed into a hash and stored on the microchip. In the last stage, verification, a new instance of the image is captured, its hash is generated and compared with the existing hash on the card chip (as in Fig. 4.7). If the two hashes are identical the person is considered to be a genuine, otherwise an impostor.

The proposed method consists in two steps: learning and hash extraction as illustrated in Fig. 4.8

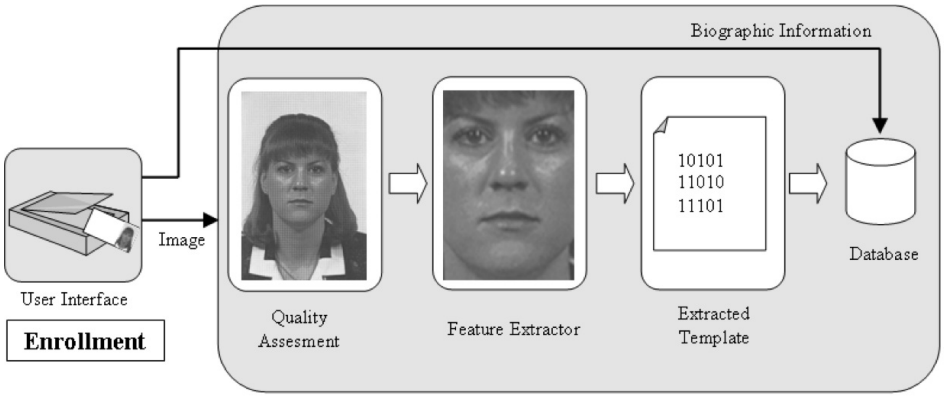


Figure 4.6: Enrollment stage.

4.3.1 Learning

In order to estimate the basis vectors the following steps are executed [Smoaca 2011]:

a) *Image preprocessing and registration.* Each image from the learning set is converted into gray tones and normalized in order to have zero mean and variance equal to one. The registration is accomplished by using a set of reference points. Each image is centered by using to coordinates of the eyes and the mouth as in Fig. 4.9.

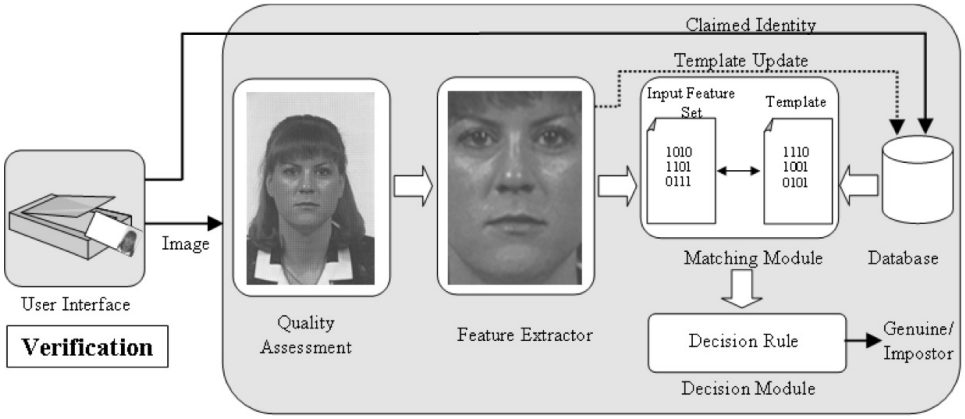


Figure 4.7: Verification stage.

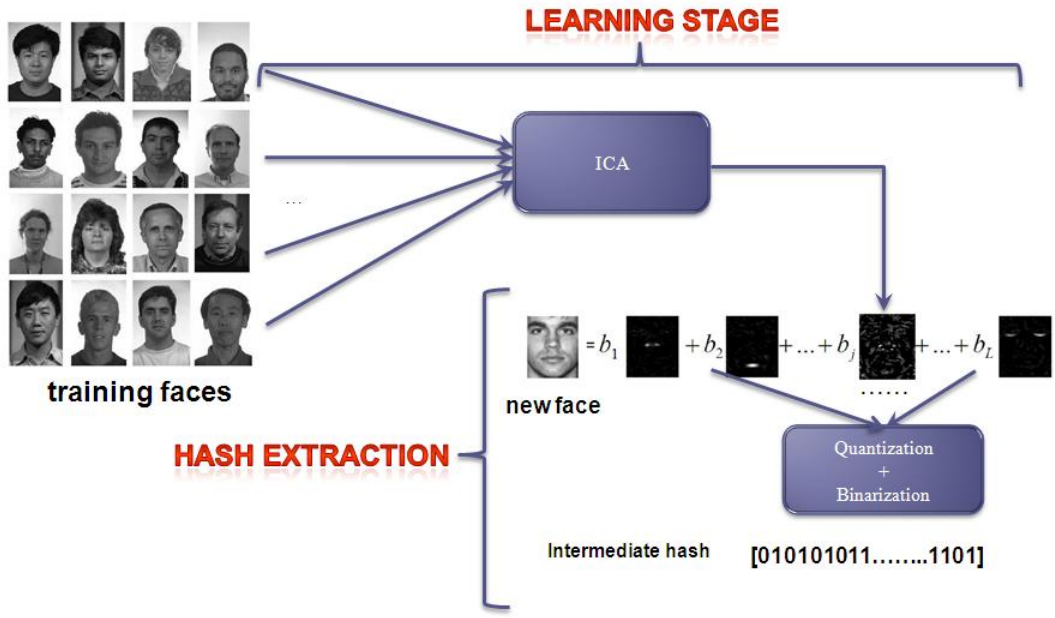


Figure 4.8: Proposed algorithm scheme: learning and hash extraction stage.

b) *Face cropping and resizing.* Each face is cropped and resized at 60x50 pixels.

c) *Image serialization.* The images are serialized and placed on the rows, respectively columns, of matrix X, depending whether ICA architecture I or ICA architecture II is employed.

d) *Basis learning.* The basis vectors are obtained by applying ICA architecture I/ICA architecture II. In architecture I the basis image vectors are independent localized features, while in architecture II they are non–independent global features.

e) *Subspace selection.* This step is necessary in order to retain only the most relevant

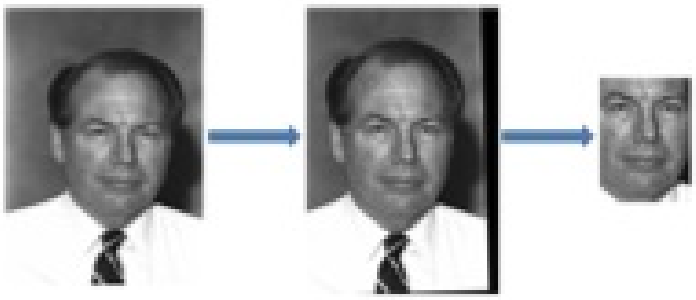


Figure 4.9: Image registration, cropping and resizing.

basis vectors from an information point of view. The subspace selection strategies described in section 4.4 are used.

4.3.2 Hash Extraction

In both enrollment and verification stage, the hash extraction step is performed. After the image capture process, any new image is first pre–processed through steps a)–c) from the learning stage. Next, the following steps are applied:

f) *Subspace projection coefficients.* The coefficients used for representing any new ID image are acquired by projecting the input image on the basis obtained in step *e)*. According to architecture I, each image I is represented as in 4.8:

$$I = \sum_{j=1}^{l} b_j s_j \tag{4.8}$$

where s_j are the independent basis vectors (ICs) and b_j the coordinates used for ICA representation.

Similarly, in architecture II, each image I is represented by:

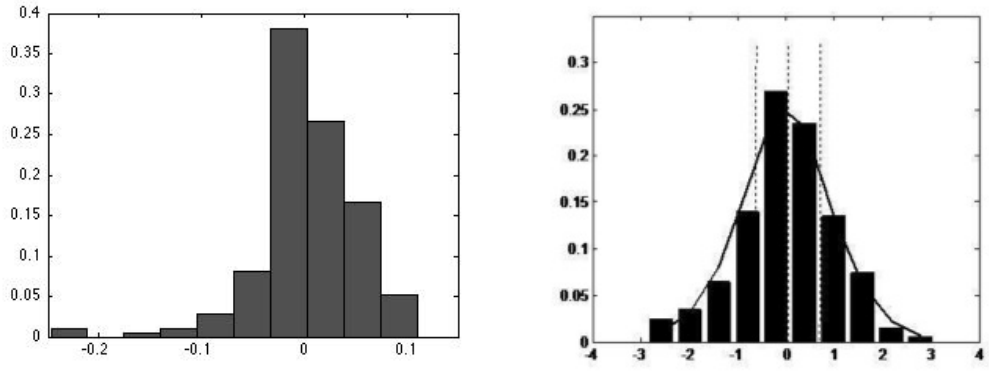
$$I = \sum_{j=1}^{l} u_j a_j \tag{4.9}$$

where a_j are the basis vectors and u_j the independent coordinates used for ICA representation (ICs).

g) *Coefficients quantization and binarization.* Since the ICA representation of a new ID image consists in continuous values, these values have to be quantized and binarized in order to obtain a binary hash.

1) *Quantization*

In the approach based on ICA architecture I, the histograms of the ICA representations for various face images cannot be modeled by a specific distribution



(a) (b)

Figure 4.10: Coefficient quantization: a) uniform quantization; b) equiprobable quantization.

(Fig. 4.10 a)). Thus we employ the uniform quantization in order to partition uniformly the interval of the subspace projection coefficients. In the ICA architecture II approach, the histograms of the independent coefficients can be modeled by a GGD as in Fig. 4.10 b) and thus an equiprobable quantization can be used to divide the subspace projection coefficients in intervals with equal probability.

The difference between the two types of quantizations is that in the uniform quantization the partition is based on the interval length of the ICA representation, whereas in the equiprobable quantization it is based on the probability density function of the ICA representation.

A. *Uniform quantization*

Let $b \in [b_{min}, b_{max}]$ denote the input of the quantizer. b_{max} and b_{min} represent the maximum and the minimum values of the variable b. A Uniform Quantizer approximates variable b according to the following rule:

$$Q_U(b) = i, \quad if \quad b \in (d_i, d_{i+1}) \tag{4.10}$$

where $i = \overline{0, ..., L-1}$ is the quantization level, L is the number of quantization levels and $d_k, k = \overline{0, ..., L-1}$ is the decision level which satisfies:

$$d_{k+1} - d_k = \Delta \tag{4.11}$$

The step size Δ of the quantizer satisfies relation 4.12:

$$\Delta = \frac{b_{max} - b_{min}}{L} \tag{4.12}$$

B. *Equiprobable quantization*

In ICA architecture II the ICs histograms can by modeled by a GGD expressed as:

$$p(x) = \frac{\beta}{2\alpha\Gamma(\frac{1}{\beta})} \exp -(\frac{|x|}{\alpha})^{\beta} \tag{4.13}$$

This allows to design an Equiprobable Quantizer described by the following rule:

$$Q_E(b) = i, \quad if \quad b \in (d_i, d_{i+1}) \tag{4.14}$$

for $i = \overline{0, ..., L-1}$.
The decision levels d_{i+1} and d_i satisfy

$$\int_{d_i}^{d_{i+1}} p(x)dx = \frac{1}{L} \tag{4.15}$$

meaning that the probability that b falls into any of the intervals $\{[d_i, d_{i+1}]$, $i = \overline{0, L-1}\}$ is $1/L$.

After quantization, the hash associated to any new image has the following form:

$$ICA\ representation = (q_1, q_2, ..., q_l) \tag{4.16}$$

where l is the number of projection coefficients and $q \in [0, 1, ..., L-1]$.

2) *Binarization*

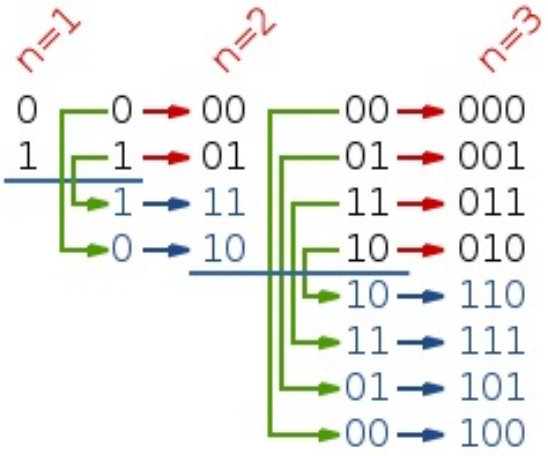


Figure 4.11: Gray code for n = 1, 2, 3 bits.

In both approaches, the discrete – binary conversion is done by applying the Gray code [Gray 1953]. An important property of the Gray code is that adjacent words in the code sequence differ in one bit only [Doran 2007] (Fig. 4.11).

After binarization, the intermediate hash has the form:

$$ICA\ representation = (g_1^n, g_2^n, ..., g_l^n) \tag{4.17}$$

where g_k^n for $k = \overline{1, l}$ refers to the Gray codeword of the discrete number k - *1* on n bits.

In the rest of the section I will refer to the intermediate hash of an image I as *h(I)*. The last stage of the enrollment and verification process consists in storing the intermediate hash in the former process and, respectively, comparing it with the pre–existing one in the latter.

4.4 Subspace selection criteria

In the print–and–scan process an image is converted from digital to hard–copy and again to digital. These conversions represent noise which modifies the content of the image and thus the extracted hash. Since the extracted hash is obtained by projecting an image on a certain subspace, it depends on the subspace chosen for image projection. This implies the importance of ICs selection in the learning stage. By choosing the most significant components for the projection subspace, the noise introduced in the print–and–scan process of an image can be attenuated.

The ICs selection can be done by the strategies presented in [Smoaca 2011]:

a) selection by PCA.

b) selection by entropic criterion.

c) combination of a) and b).

The selection by PCA can be used in both ICA architecture I and II approaches, while the selection by entropic criterion as presented in [Smoaca 2011] is used only for the former one.

4.4.1 Selection by PCA

The selection by PCA is done by using a variance criterion since it takes into consideration only second order statistics. The removal of components is accomplished in the pre–processing step of the ICA process. It consists in retaining only the highest variance coefficients. An example is illustrated in Fig. 4.12. Using the subspace obtained after PCA selection in the ICA process the performance is improved and the complexity is reduced.

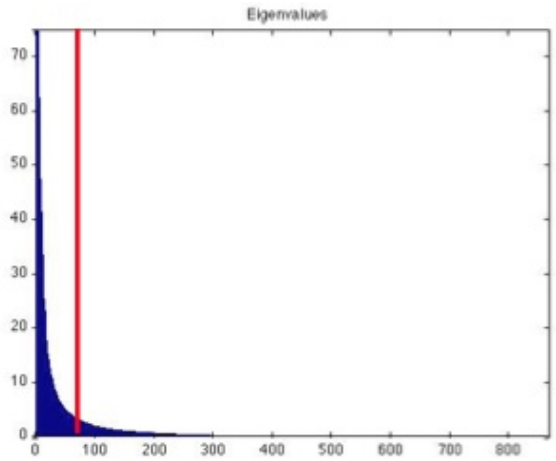


Figure 4.12: Selection of the eigenvalues: only 85% of the original signal energy (left side of the red bar) are retained.

4.4.2 Entropic criterion

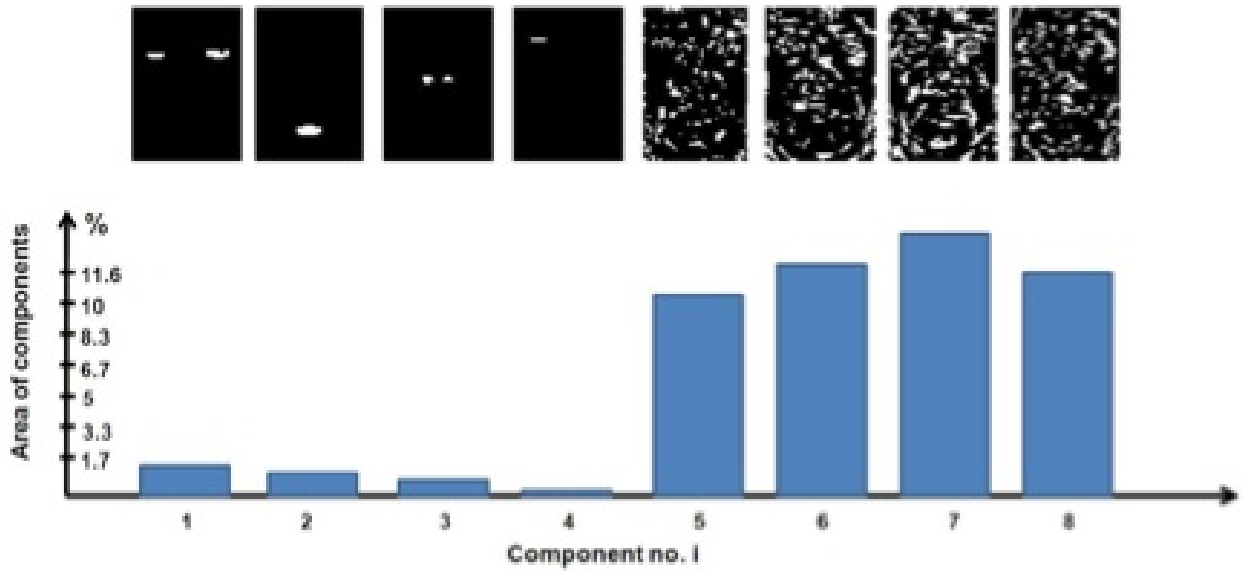


Figure 4.13: Proposed selection of the components by area–criterion. The top row shows eight binarized ICs obtained with ICA architecture I. The second row displays the area of the above ICs. Only ICs with low area must be retained.

The goal is to obtain an adequate projection subspace which, when projecting an image that has been affected by the scan noise introduced when authenticating the ID document, will lead with high probability to the same hash value as the one stored on the microchip of the document. For the proposed image hashing scheme, a suitable projection subspace can be obtained by selecting only the components that are relevant in the sense of information content.

In Fig. 4.13 eight binarized ICs obtained by performing ICA architecture I are illustrated. It is interesting to notice that, even if by using architecture I the basis vectors should be localized features, there are several components in which, behind the human biometric features (eyes, mouth, nose, etc.), more or less salient face images appear. This means that the relevant components in the sense of information content are the ones with less salient face images appearing behind face features.

Since the basis vectors used for image projection are obtained by performing FastICA algorithm, which maximizes the negentropy, the choice of an entropic criterion to select the ICs is a natural one. Two types of entropic selection are proposed in this section:

- *global entropic selection*: keeps the most significant components with minimal entropy;

- *local entropic selection*: keeps the most significant components with maximal entropy;

There are many methods for entropy estimation. For the entropic criterion there is no need for an accurate source model for face features. The goal here is rather to have an efficient and fast selection than an accurate model. This is the reason why face features binarization and zero-order entropy estimate can be considered. It is known that for a binary memory–less source of information, the entropy is maximum for $p = 0.5$ and symmetrical about this point (p is the probability of one of the symbols).

Entropy estimate can be advantageously replaced by area measurements. On the second row of Fig. 4.13, the area values corresponding to eight binarized ICs are shown. It can be observed that the relevant components, with low entropy, have low area value.

The global entropic criterion is presented in Fig. 4.14. The global criterion considers only the components that present low entropy as relevant. In this case low entropy ICs are the ICs with less salient face images appearing behind face features. Using this space selection criterion, only the components with low area value are retained and used further in the projection step. The area is computed on the binarized components as the ratio between the white pixels and black pixels. The threshold employed for binarization has been obtained by applying Otsu's method on the ICs.

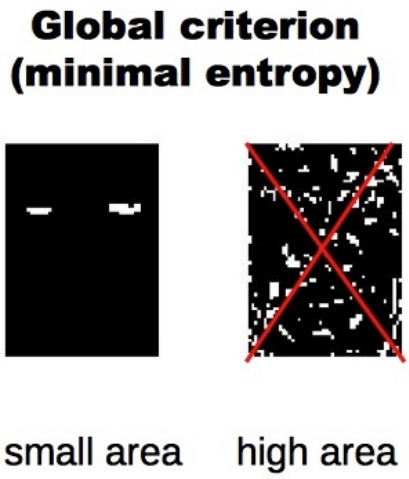


Figure 4.14: Global entropic criterion.

Further, we refined the components selection, by discarding those with no salient face features. In this strategy, after binarizing the components as before, the ICs images have been split into 3 strips as shown in Fig. 4.15: the upper third containing the eyes and eyebrows, the middle third containing the nose and the lower one containing the mouth. The most significant ICs are the ones that have maximal local entropy, which implies that the ICs have maximum entropy (bigger area) on a single strip and (quasi) null entropy on the other two. By using this strategy, the projection subspace can be constructed so that it will contain only one type of face feature, i.e. eyes, but it is preferable to have all types of face features, i. e. eyes, mouth, nose, eyebrows.

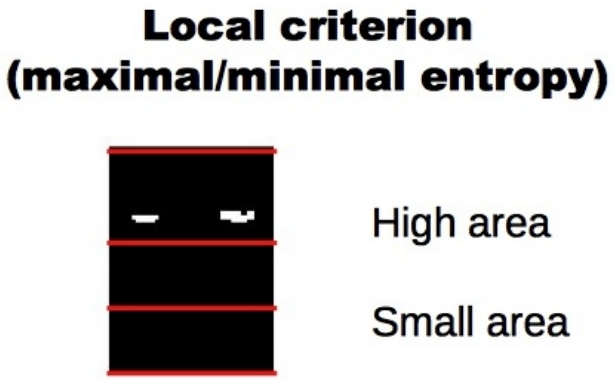


Figure 4.15: Local entropic criterion

4.5 Results on simulated data

Our method was tested on a subset of the FERET database of facial images ([Phillips 1998], [Phillips 2000]). In order to stay close to ID images, only the subjects with frontal view and neutral expression subjects were kept. In the learning stage a training set of 300 gray level images of 384x256 pixels (Fig. 4.16) has been used to determine the basis vectors in both ICA architecture I and II approaches. For enrollment and verification, another

Figure 4.16: Samples from FERET database.

set of 210 different subjects' images was considered. All the input ID images were passed through steps a)–b) from the learning stage. They were registrated, cropped and resized to 60x50 pixels as shown in Fig. 4.9. All the images, used for learning, enrollment and

verification, were registered to a reference set of coordinates obtained by averaging the eyes and mouth coordinates provided by the FERET database .

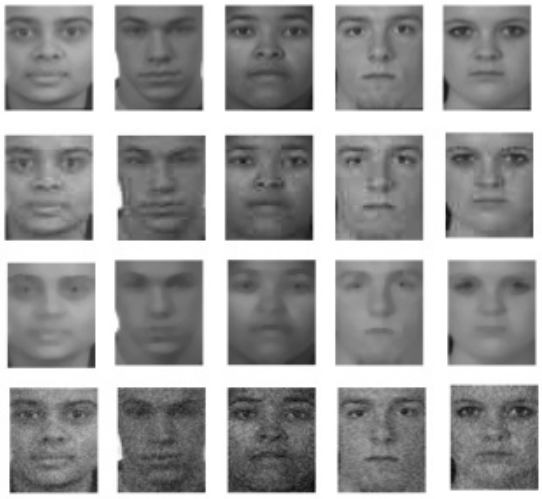


Figure 4.17: The experimental data. Examples of images affected by affine transforms on the first row, JPEG compression on the second row, median filtering on the third and addition of white gaussian noise on the last row.

By projecting the input ID images used for enrollment and verification on the learned basis vectors and by quantizing the obtained coordinates on 8 levels and Gray encoding, an intermediate binary hash *h(I)* is obtained for each image *I*.

Since the designed hashing algorithm must be robust to print–and–scan attack which is considered as a combination of several attacks, such as filtering, noise adding, scaling, small affine transform, small random twist, JPEG compression, luminance and contrast adjustment and chrominance variations [Lin 1999b], the intermediate hashes *h(I)* were tested against a series of stirmark attacks [Petitcolas 1998], [Petitcolas 2000] (affine transforms, JPEG compression, additive Gaussian white noise and median filter). Samples of the attacked test images are shown in Fig. 4.17. The Hamming distance was employed to compare the binary intermediate hashes before (extracted in the enrollment stage) and after the attack (extracted in the verification stage). The performance of the method can be described by using the *r–value* and Receiver Operating Characteristic (ROC) curves. The *r–value* measures the gap between the means of the Hamming distances distributions of the similar faces (originals and attacked) and dissimilar ones: it is defined as:

$$r - value = \frac{|\mu_s - \mu_d|}{\sigma_s + \sigma_d} \tag{4.18}$$

where μ_s, σ_s and μ_d, σ_d are the mean and standard deviation of the similar faces, respectively of the dissimilar faces. The higher the $r - value$ is, the higher the performance of the method is, i.e. the gap between the two Hamming distances distributions is larger.

4.5.1 Approach ICA architecture I

For the results presented in tables 4.1 and 4.2, FastICA software was used to analyze the learning set and extract the basis vectors. For 300 input ID images, FastICA can extract a maximum of 300 ICs leading to an intermediate binary hash h of 900 bits. In order to shorten the length of h, PCA was used as a pre–processor before ICA. Thus, we retained only the first 60 coefficients carrying 94% of the signal's energy. Further, the ICs with salient features are selected by using the global entropy criterion. The 60 basis vectors were split into two subsets: one of 33 ICs with low entropy (Case I) and another one of 27 ICs with high entropy (Case II). The separation in the two subsets was done by evaluating the white area on the binarized basis vectors. For all experiments, Otsu's method was used to determine the binarization threshold.

	Attack	Similar faces		Dissimilar faces		r-value
		mean	σ	mean	σ	
Case I	AFFINE _1	5.6	2.3	37.7	6.5	3.7
	AFFINE _2	18	4.3	38.5	6.3	1.9
	AFFINE _3	6.6	2.4	37.8	6.4	3.5
	AFFINE _4	28.7	5.7	42	5.6	1.2
	AFFINE _5	8.2	2.7	37.8	6.4	3.3
	AFFINE _6	9.9	3.2	38	6.3	3
	WN $\sigma = 0.01$	17.5	5.5	34.53	7.1	1.3
	Median 5×5	13.6	4.4	38.1	6.4	2.3
Case II	AFFINE _1	4	1.9	31.7	6.7	3.2
	AFFINE _2	18.7	4	33.4	5.6	1.5
	AFFINE _3	5	2	31.9	6.5	3.1
	AFFINE _4	21.5	4.5	34.9	5.2	1.3
	AFFINE _5	7.3	2.6	31.8	6.5	2.7
	AFFINE _6	9.1	2.9	32.1	6.4	2.5
	WN $\sigma = 0.01$	10.2	3.7	24.3	7	1.3
	Median 5×5	9.2	3.1	32.7	6.8	2.3

Table 4.1: Comparison between strong features and weak features.

Table 4.1 shows the importance of selecting only relevant basis vectors in the sense of information content. The images were not %100 recognized because of the smaller number of ICs (≈ 30 in both situations), but the performance is higher in Case I. For example, for the AFFINE_1 geometric attack, the *r–value* is higher in Case I (3.7 vs. 3.2).

The other reducing strategies (only PCA, PCA and local entropy (LE) selection, PCA with local and global entropy selection combined – LE–GE) were tested in the same way. The results in table 4.2 show that the *r–value* increases with the number of selected ICs.

For example, for the geometric attack (AFFINE_3) when using the first 180 PC, the *r–value* is 4.7 compared with 4.3 obtained for the first 120 PC. When tuning the number of quantization levels, for a higher level of quantization the *r–value* increases

	Attack	r-value	Attack	r -value
180 PCA	AFFINE _1	5	AFFINE _2	2.4
$L = 8$	AFFINE _3	4.7	AFFINE _4	2.1
	AFFINE _5	4.2	AFFINE _6	3.8
	JPEG Q = 15	4.1	JPEG = 20	4
	WN $\sigma = 0.01$	2.5	WN $\sigma = 0.02$	1.9
	Median 3×3	4.4	Median 5×5	3.1
120 PCA	AFFINE _1	4.7	AFFINE _2	2.1
$L = 8$	AFFINE _3	4.3	AFFINE _4	1.8
	AFFINE _5	3.8	AFFINE _6	3.5
	JPEG Q = 15	4.1	JPEG = 20	4
	WN $\sigma = 0.01$	2.3	WN $\sigma = 0.02$	2.3
	Median 3×3	4.2	Median 5×5	3
120 PCA	AFFINE _1	3.7	AFFINE _2	2.1
$L = 4$	AFFINE _3	3.5	AFFINE _4	1.7
	AFFINE _5	3.2	AFFINE _6	3
	JPEG Q = 15	4.1	JPEG = 20	4
	WN $\sigma = 0.01$	2	WN $\sigma = 0.02$	1.7
	Median 3×3	2.5	Median 5×5	2.7
120 LE	AFFINE _1	5.7	AFFINE _2	2.2
$L = 16$	AFFINE _3	4.8	AFFINE _4	2
	AFFINE _5	4.2	AFFINE _6	4
	JPEG Q = 15	4.4	JPEG = 20	5
	WN $\sigma = 0.01$	2.5	WN $\sigma = 0.02$	1.9
	Median 3×3	4.3	Median 5×5	3

Table 4.2: Results for different attacks and selection criteria for FastICA algorithm: PC, LE (Local Entropy).

(2.1 vs. 1.7 for AFFINE_4 with 8, respectively 4 levels). For a constant number of components and quantization levels (120, respectively 8), the *r–value* is higher when using an entropic criterion (for AFFINE_5, the *r–value* is 4 for 120 LE and 3.8 for 120 PCA). The same tendency was observed also for the attacks where not all the images were correctly identified.

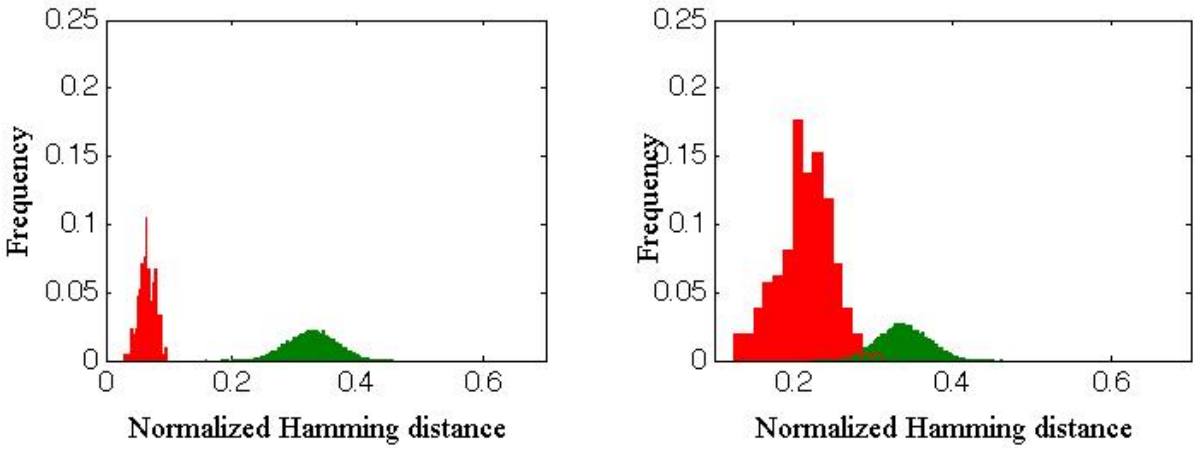
Next, a second step is performed on the 120 LE components, by selecting the basis vectors with global low area. For the cases where full recognition is not reached, more ID photos are correctly identified than when using only PC selection. For example, for AFFINE_4 attack, the $r - value$ is 1.5 for 66PCA and 1.9 for 66LE–GE.

The distribution of the normalized Hamming distances for the various attacks from Fig. 4.17 are shown in Fig. 4.18. The genuine and impostors distributions are well separated for geometric attacks like AFFINE_1 and AFFINE_3, for JPEG compression with a quality factor of 15 or higher, for median filtering by a 3 x 3 window and for white gaussian noise with $\sigma = 0.01$. The distributions overlap in the case of geometric attacks

	Attack	r-value	Attack	r -value
66 PCA	AFFINE _1	4.8	AFFINE _2	1.7
$L = 8$	AFFINE _3	4.3	AFFINE _4	1.5
	AFFINE _5	3.7	AFFINE _6	3.4
	JPEG Q = 15	4.1	JPEG = 20	4
	WN $\sigma = 0.01$	2.3	WN $\sigma = 0.02$	1.7
	Median 3×3	4.1	Median 5×5	2.8
66 LE – GE	AFFINE _1	4.4	AFFINE _2	1.9
$L = 8$	AFFINE _3	4.1	AFFINE _4	1.9
	AFFINE _5	3.5	AFFINE _6	3.3
	JPEG Q = 15	4.1	JPEG = 20	4
	WN $\sigma = 0.01$	5	WN $\sigma = 0.02$	4
	Median 3×3	3.9	Median 5×5	2.6

Table 4.3: Results for different attacks and selection criteria for FastICA algorithm: PC, LE – GE.

like AFFINE_2 and AFFINE_4, median filtering by a 5 x 5 window or higher and for white gaussian noise with $\sigma = 0.02$ or higher.



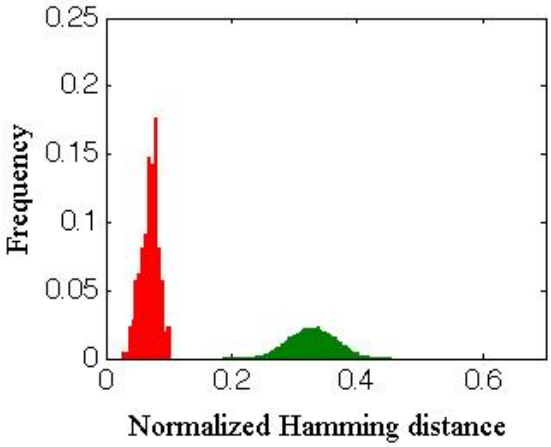
(a) AFFINE_1, $r - value = 5.7$ (b) AFFINE_2,$r - value = 2.2$

Figure 4.18

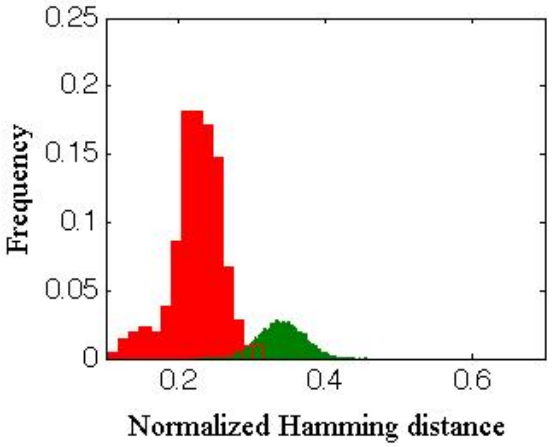
In Fig. 4.19 the histograms for a constant number of ICs in the case of JPEG compression are shown. The more quantization levels, the more separated the genuine and impostors distributions are.

Fig. 4.20 presents the distribution of genuine and impostors for the geometric attack AFFINE_2 when the number of quantization levels is kept constant. The two histograms are well separated for a higher number of ICs.

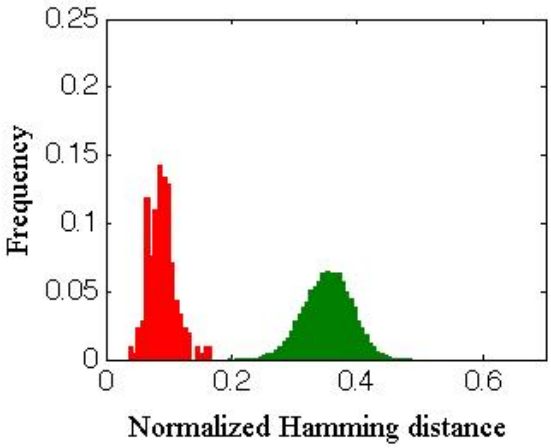
The influence of various ICA algorithms on the genuine and impostors distributions is illustrated in Fig. 4.21. It is with the FastICA algorithm that the two histograms are



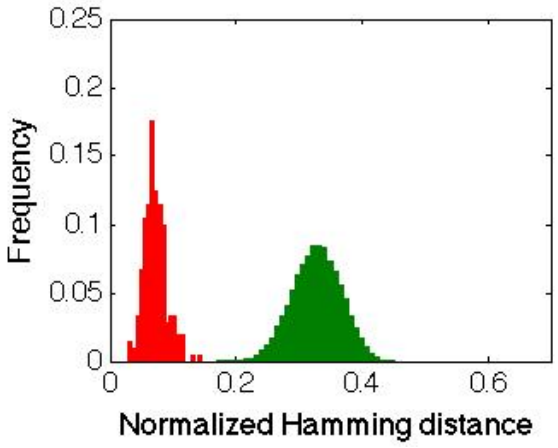
(c) AFFINE_3,$r - value = 4.8$



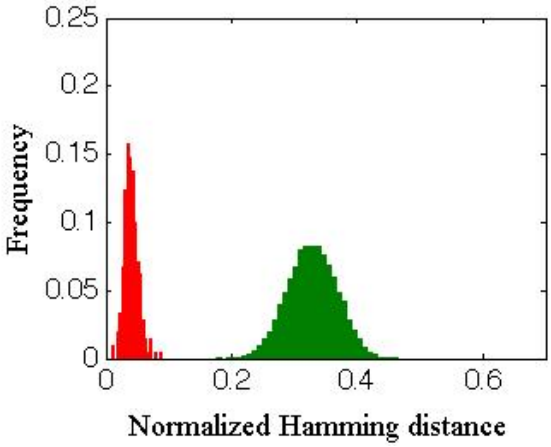
(d) AFFINE_4,$r - value = 2.4$



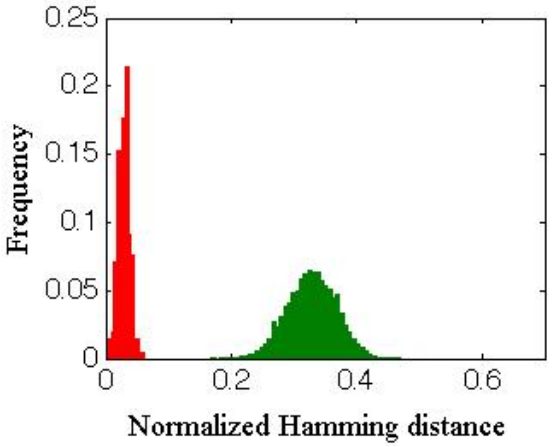
(e) JPEG compression with Q = 15,$r - value = 4.4$



(f) JPEG compression with Q = 20,$r - value = 5$

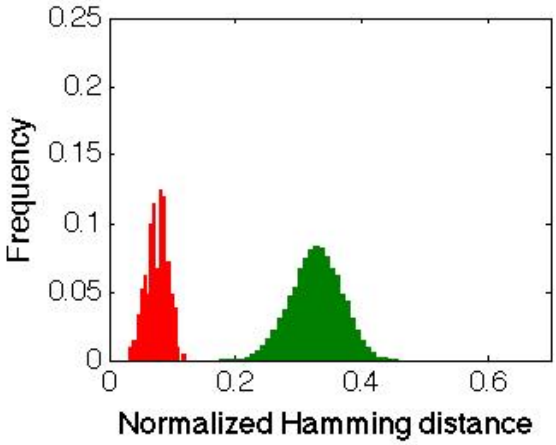


(g) JPEG compression with Q = 50,$r - value = 6.4$

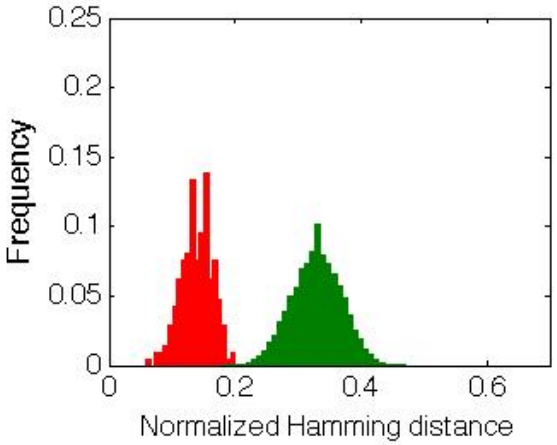


(h) JPEG compression with Q = 70,$r - value = 7.4$

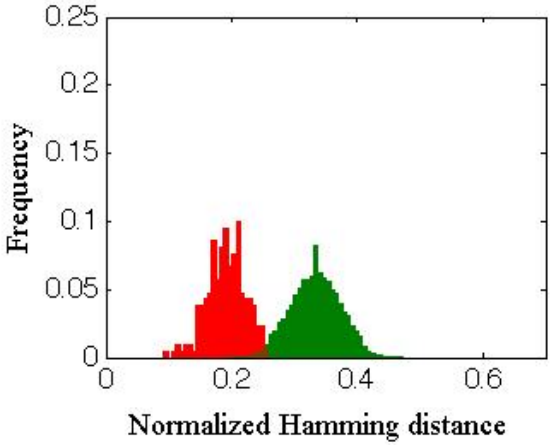
Figure 4.18



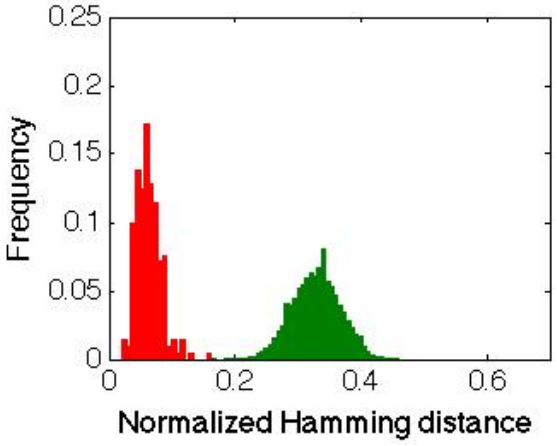
(i) 3 x 3 median filter,$r - value = 4.7$



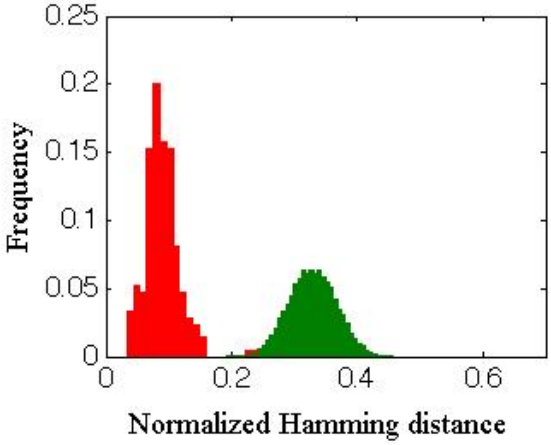
(j) 5 x 5 median filter,$r - value = 3$



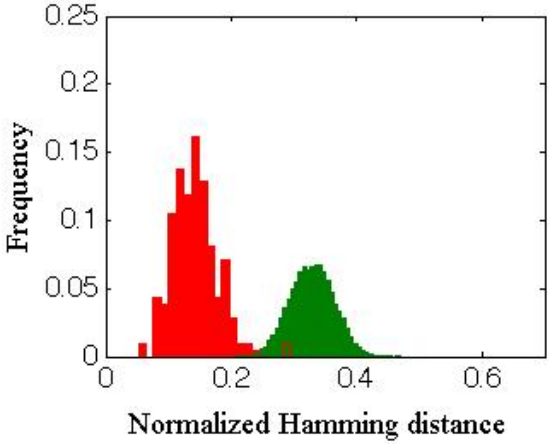
(k) 7 x 7 median filter,$r - value = 2.1$



(l) gaussian noise with $\sigma = 0.01$,$r - value = 5$

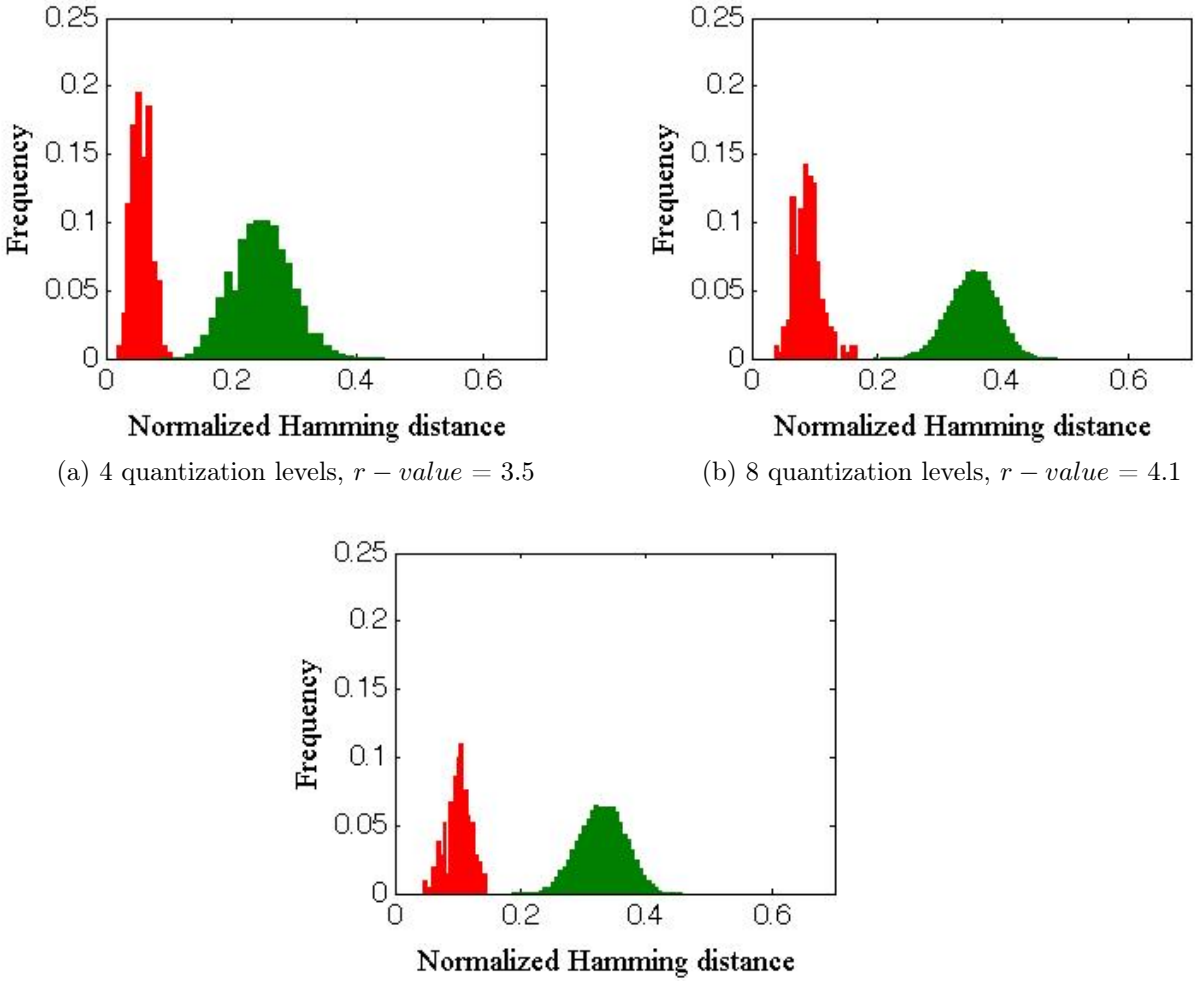


(m) gaussian noise with $\sigma = 0.02$,$r - value = 4$



(n) gaussian noise with $\sigma = 0.05$,$r - value = 3$

Figure 4.18: Authentic and impostors distributions for several attacks for 120 LE and L = 16.



(a) 4 quantization levels, $r - value = 3.5$

(b) 8 quantization levels, $r - value = 4.1$

(c) 16 quantization levels, $r - value = 4.4$

Figure 4.19: Genuine and impostors histograms in the case of JPEG compression attack with 15 quality factor and 120 LE with different quantization level.

the most separated.

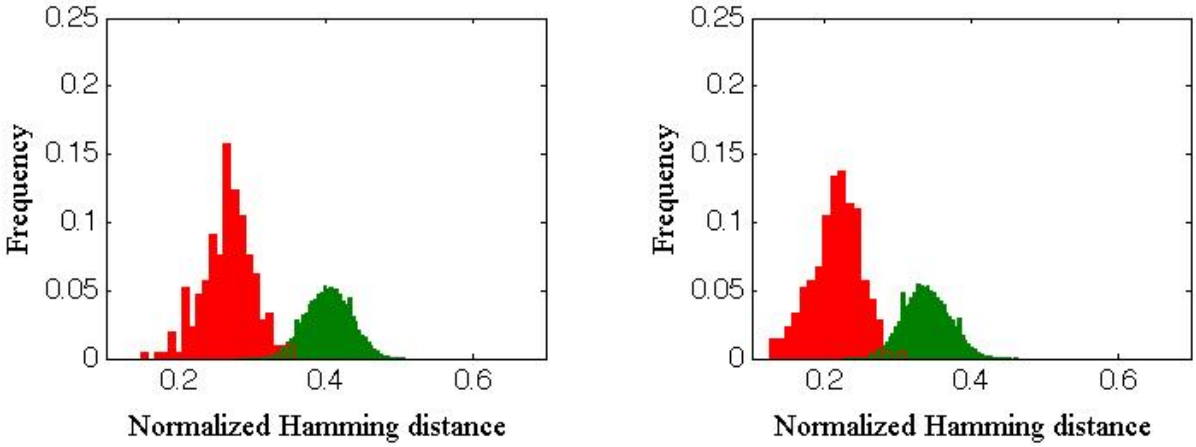
The two distributions shown in Fig. 4.18 – 4.24 are not always well separated. The overlap of the distributions gives two types of errors known in the statistical decision theory as :

1) mistaking features vectors from two different persons to be from the same person (called false match or false acceptance rate (FAR)).

2) mistaking two features vectors from the same person to be from two different persons (called false non–match or false rejection rate (FRR)).

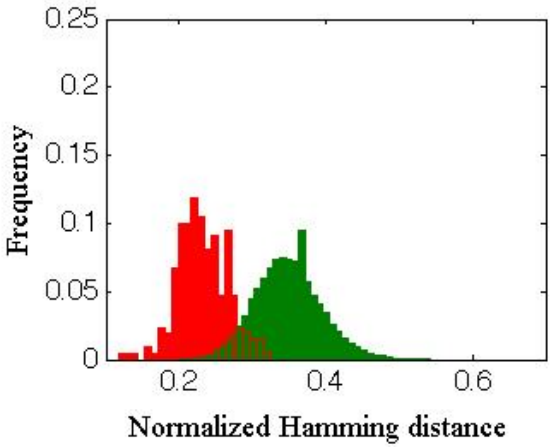
These errors may be estimated from the experimental data.

Mathematically, these two types of errors are usually expressed as the following con-



(a) 180 ICs, $r - value = 2.2$

(b) 120 ICs, $r - value = 2$



(c) 60 ICs, $r - value = 1.8$

Figure 4.20: Genuine and impostors distribution in the case of AFFINE_2 attack for different number of ICs and $L = 16$.

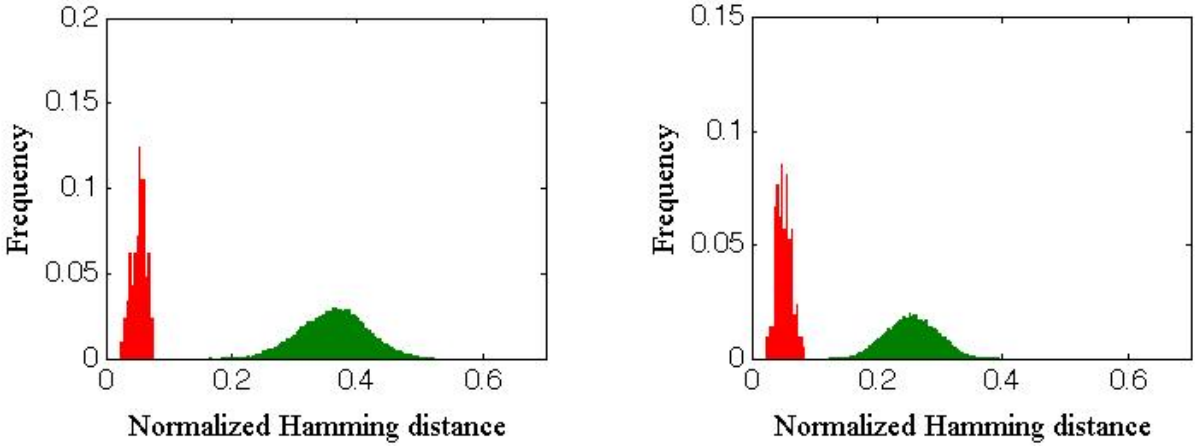
ditional probabilities:

$$FAR = P(D_0|H_1) \tag{4.19}$$

$$FRR = P(\bar{D}_0|H_0) \tag{4.20}$$

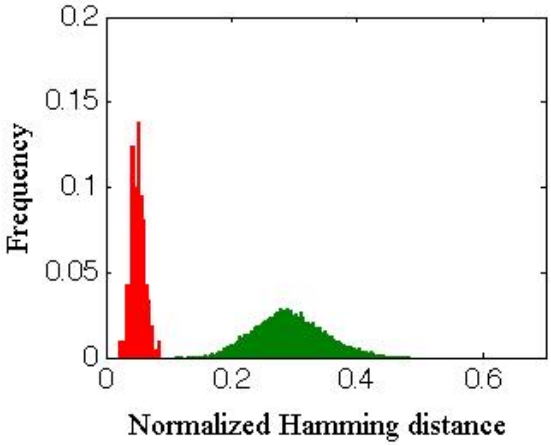
where H_0 is the null hypothesis, i.e. the extracted features vector corresponds to the reference one (one–to–one comparison), H_1 is the alternative hypothesis and D_0 is the decision region associated to H_0, i.e. the Hamming distance lower than a predefined threshold d_0 (ID photograph is genuine). $\bar{D}_0$ is the complementary of the decision region D_0.

In order to determine a relevant configuration (ICs number, quantization levels, subspace selection criterion, ICA algorithm), we used also ROC curve analysis [Fawcett 2006]. The ROC curve is obtained as genuine acceptance rate (GAR) (1-FRR) vs. FAR when varying d_0.

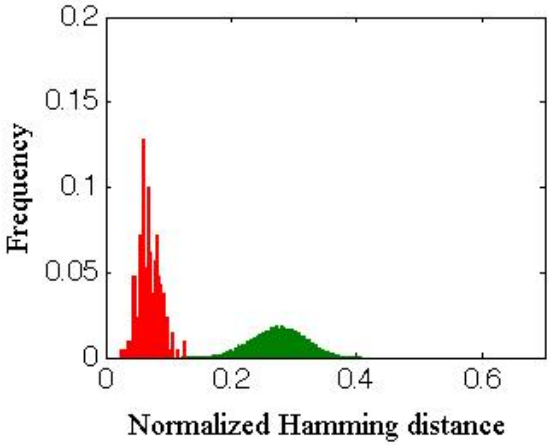


(a) FastICA, $r - value = 5.7$

(b) InfoMax, $r - value = 4$



(c) Pearson, $r - value = 4.7$



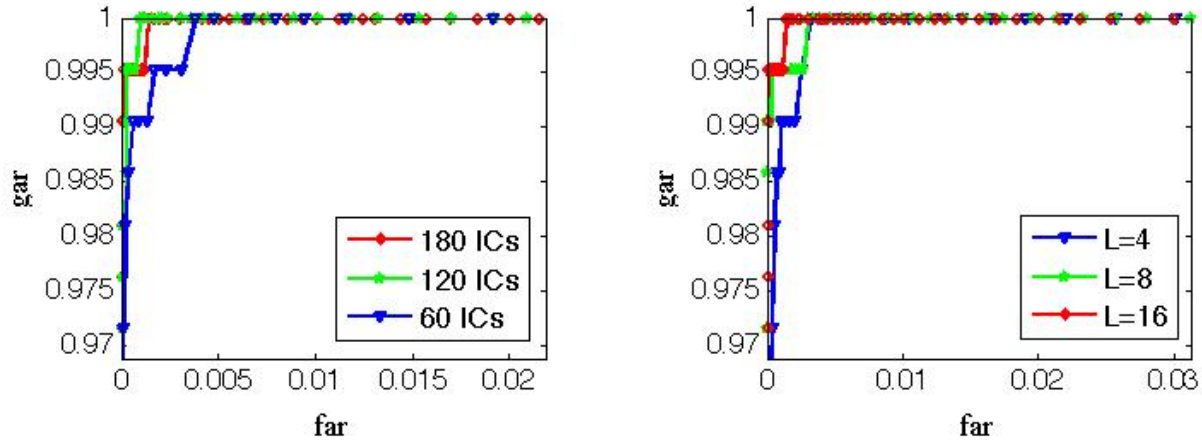
(d) ERICA, $r - value = 3.8$

Figure 4.21: Genuine and impostors histograms in the case of AFFINE_1 attack and various ICA algorithms for 180 ICs and $L = 16$.

Examples of ROC curves are shown in Fig. 4.22–4.24. Fig. 4.22a) compares the performances of the median filtering attack for a constant number of quantization levels and different number of ICs. As for the $r - value$, the performance increases with the number of ICs. It is visible that the curve for 180 components is superior to the other ones. In Fig. 4.22b), the ROC curves for different quantization levels are illustrated. For a GAR value of 99.5% the smaller FAR value is obtained for $L = 16$ quantization levels (0.009% vs. 0.03%, 0.25%).

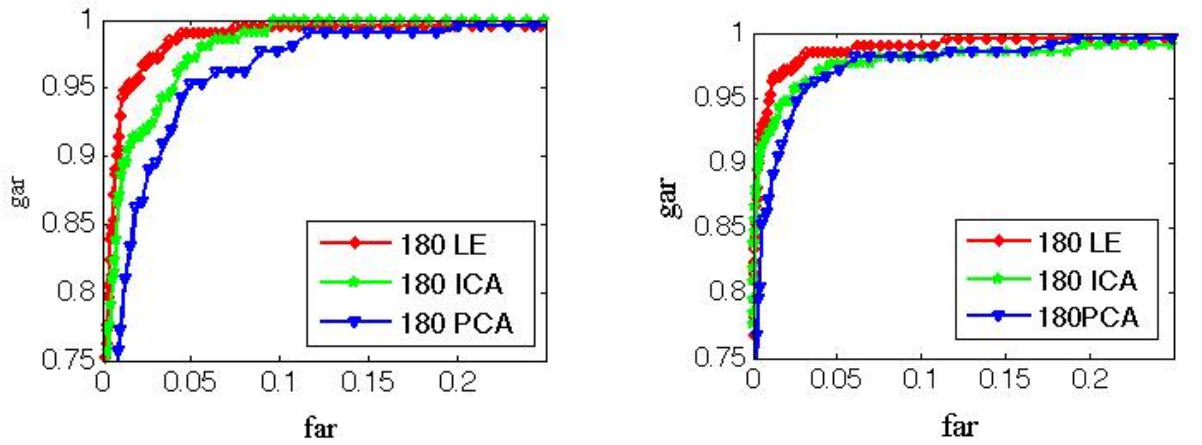
Fig. 4.24 compares different subspace reducing strategies in the case of median filtering and white gaussian noise addition. The local entropic criterion achieves a low FAR for a high GAR.

The same tendency observed for the $r - value$ and the ROC curves is valid also when employing another ICA algorithm.



(a) (b)

Figure 4.22: ROC curves for median filtering attack with a 5 x 5 window for: a) $L = 16$ and different number of ICs; and b) 180 ICs and different quantization levels.



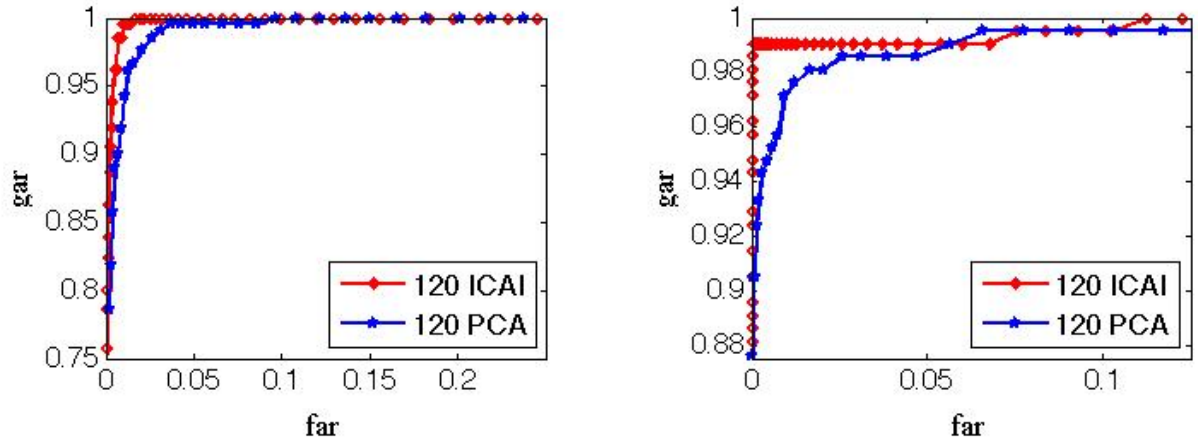
(a) (b)

Figure 4.23: ROC curves for different subspace strategies in the case of: a) median filter with a 7 x 7 window; b) gaussian white noise with $\sigma = 0.05$.

4.5.2 Approach ICA architecture II

For the experiments in this section we have extracted the basis vectors from the same learning set employed in the above section. In this approach, since the image pixels are considered variables, FastICA can extract a maximum of 3000 ICs. PCA is used as a pre–processor selecting the components that conserve 98.1% of the energy of the original signal.

The results in table 4.4 reveal that when tuning the number of Ics, the performance is better or equal for a higher number. For example, for AFFINE_5 r is 3.9 for 180 ICs and 3.7 for 120 ICs. Only AFFINE_3 and AFFINE_7 have a higher *r–value* (4.5



(a) (b)

Figure 4.24: ROC curves for PCA and ICAI in the case of: a) median filter with a 7 x 7 window; b) gaussian white noise with $\sigma = 0.05$.

vs. 4.3 and 3.6 vs. 3.4). For a higher number of quantization level, the algorithm has worse performance. For a smaller quantization level the *r–value* is higher or equal (for AFFINE_4 1.1 vs. 1.2 for 8, respectively 4 quantization levels).

The same tendency is valid also when using InfoMax, Pearson and ERICA algorithms to extract the ICs. The best performance is achieved when using FastICA algorithm.

The genuine and impostors distributions for a constant number of IC in the case of the JPEG compression are shown in Fig. 4.26. The two histograms are not as well separated as the ones in Fig. 4.19. An inverse tendency is noticed. The smaller the quantization level is, the better separated the two histograms are.

Fig. 4.27 shows the two distributions in the case of the geometric attack AFFINE_2. The number of ICs are tuned and the quantization level is kept constant. For a higher number of ICs the overlapping interval is smaller.

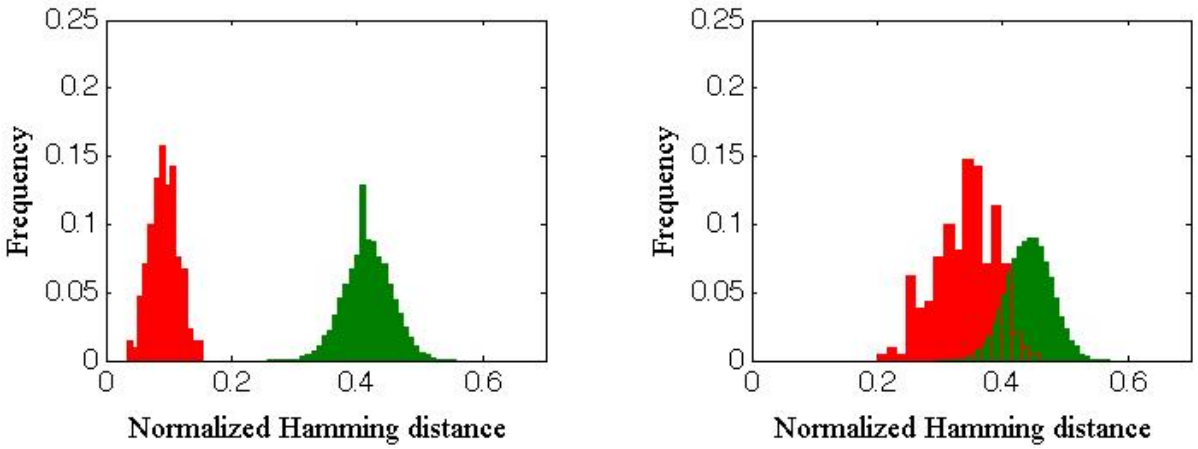
Fig. 4.28 shows the ROC curves for the median filtering attack when tuning the number of ICs and quantization levels. In Fig. 4.28a), the quantization level is kept constant and the number of ICs is tuned. A better ROC curve is obtained for a higher number of ICs. Fig. 4.28b) shows the ROC curves for a constant number of ICs. For a GAR of 99.5% the smaller FAR is obtained for $L = 4$ quantization levels (0.12% vs. 0.27%, 0.3%).

The ROC curves, comparing the performances of different subspace selection strategies for median filtering and white gaussian noise, are illustrated in Fig. 4.29. The ROC curve for the 180 components obtained by applying ICA is superior to the one obtained by applying PCA only.

When comparing the two architectures, ICA architecture I has better results in the cases where full recognition is not achieved, i.e. when the two distributions overlap, but also when all the images are correctly recognized. For example, for the geometric attack

	Attack	r–value	Attack	r–value
180 PC	AFFINE _1	5.8	AFFINE _2	1.2
$L = 8$	AFFINE _3	4.3	AFFINE _4	1.1
	AFFINE _5	3.9	AFFINE _6	2.8
	JPEG Q=15	4.1	JPEG Q=20	3.1
	WN $\sigma = 0.01$	2.5	WN $\sigma = 0.02$	1.9
	Median 3×3	4.6	Median 5×5	2.4
120 PC	AFFINE _1	5.6	AFFINE _2	1.2
$L = 8$	AFFINE _3	4.5	AFFINE _4	1.1
	AFFINE _5	3.7	AFFINE _6	2.8
	JPEG Q=15	4.1	JPEG Q=20	3.1
	WN $\sigma = 0.01$	2.3	WN $\sigma = 0.02$	2.3
	Median 3×3	4.6	Median 5×5	2.4
120 PC	AFFINE _1	5.4	AFFINE _2	1.2
$L = 4$	AFFINE _3	4.7	AFFINE _4	1.2
	AFFINE _5	4	AFFINE _6	3.2
	JPEG Q=15	4.1	JPEG Q=20	4.8
	WN $\sigma = 0.01$	5.2	WN $\sigma = 0.02$	4.3
	Median 3×3	4.6	Median 5×5	2.4

Table 4.4: Results for different attacks and selection criteria for FastICA.

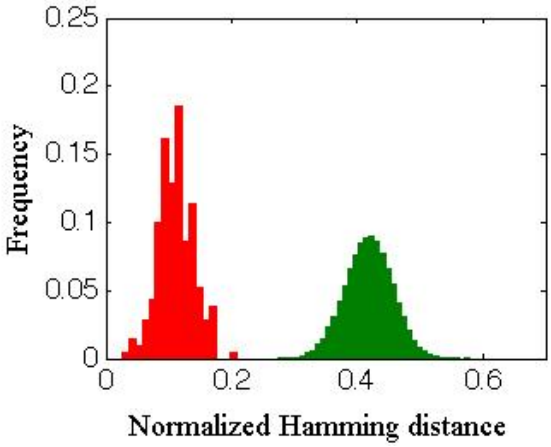


(a) AFFINE_1,$r - value = 5.4$ (b) AFFINE_2,$r - value = 1.2$

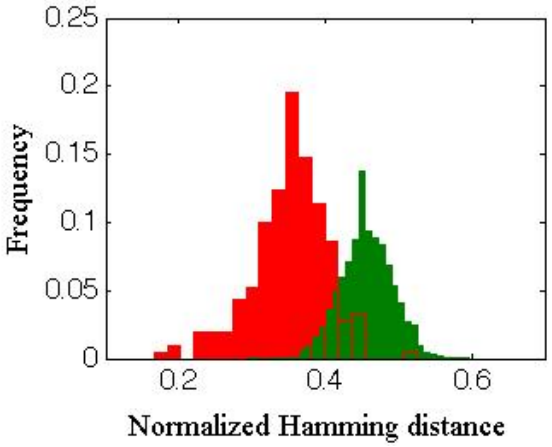
Figure 4.25

AFFINE_2, a $r - value$ of 2.2 is obtained when employing ICA architecture I compared to 1.2 when employing ICA architecture II.

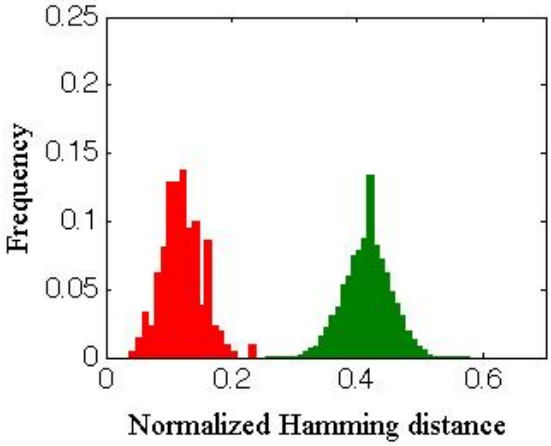
Examples of ROC curves obtained for ICA I and ICA II approaches and in the case of AFFINE transforms, median filtering and noise addition are shown in Fig. 4.30. The ROC curves obtained by using ICA I are superior to the ones obtained by using ICA II for the selected attacks.



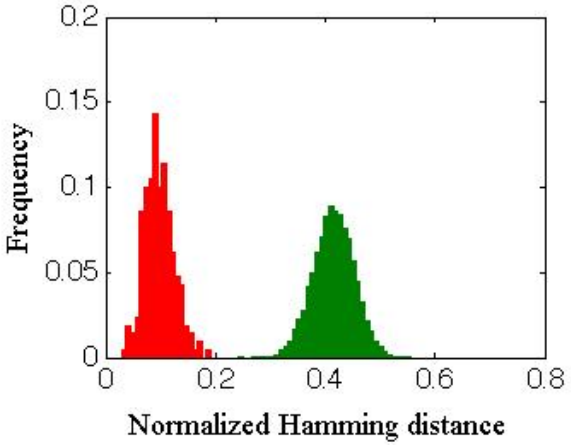
(c) AFFINE_3,$r - value = 4.6$



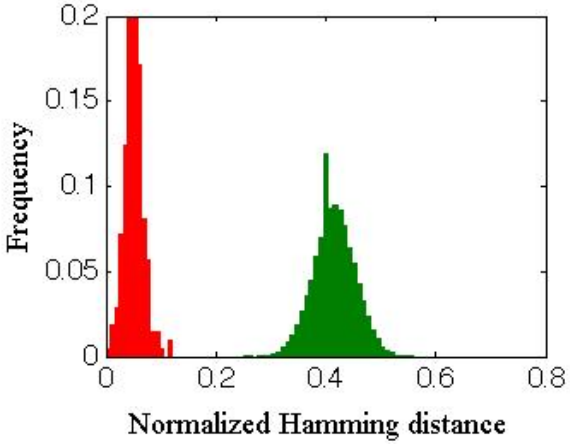
(d) AFFINE_4,$r - value = 1.2$



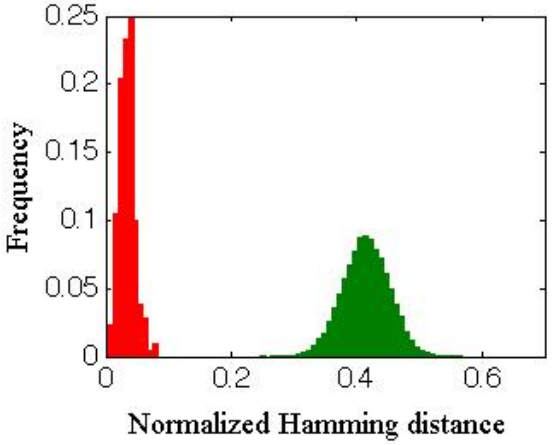
(e) JPEG compression with Q = 15,$r - value = 4.1$



(f) JPEG compression with Q = 20,$r - value = 4.8$

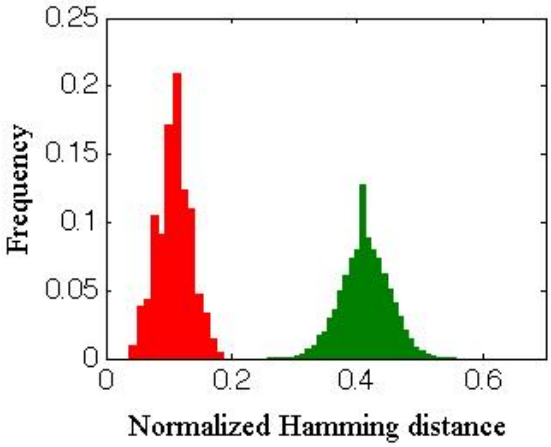


(g) JPEG compression with Q = 50,$r - value = 6.4$

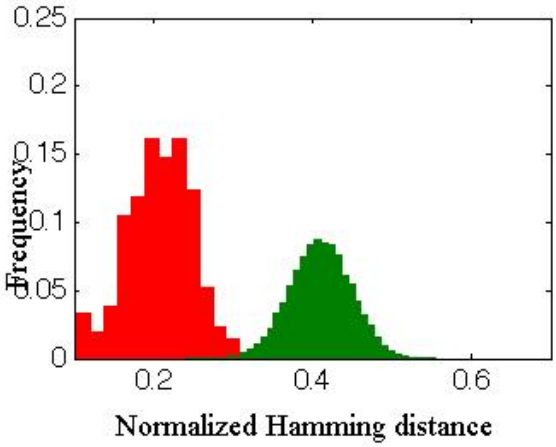


(h) JPEG compression with Q = 70,$r - value = 7.4$

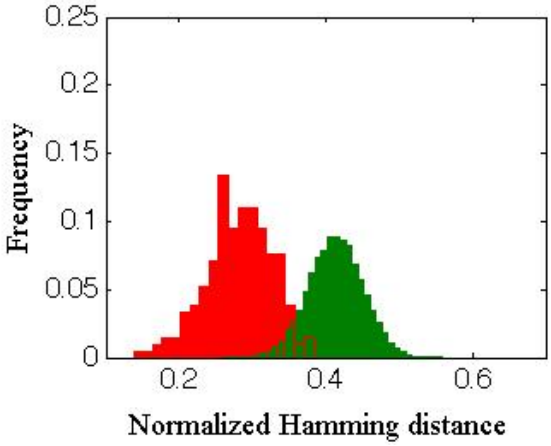
Figure 4.25



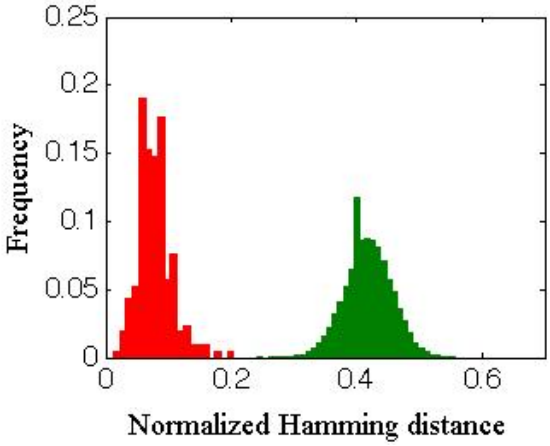
(i) 3 x 3 median filter,$r - value = 4.6$



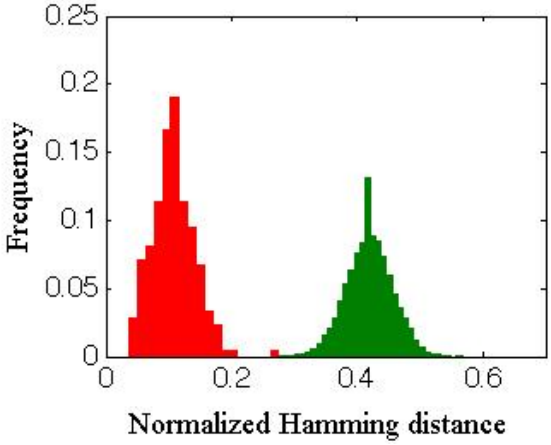
(j) 5 x 5 median filter,$r - value = 2.4$



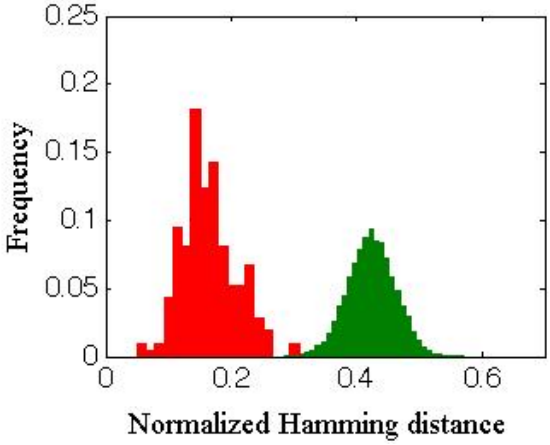
(k) 7 x 7 median filter$r - value = 1.6$



(l) gaussian noise with $\sigma = 0.01$,$r - value = 5.2$

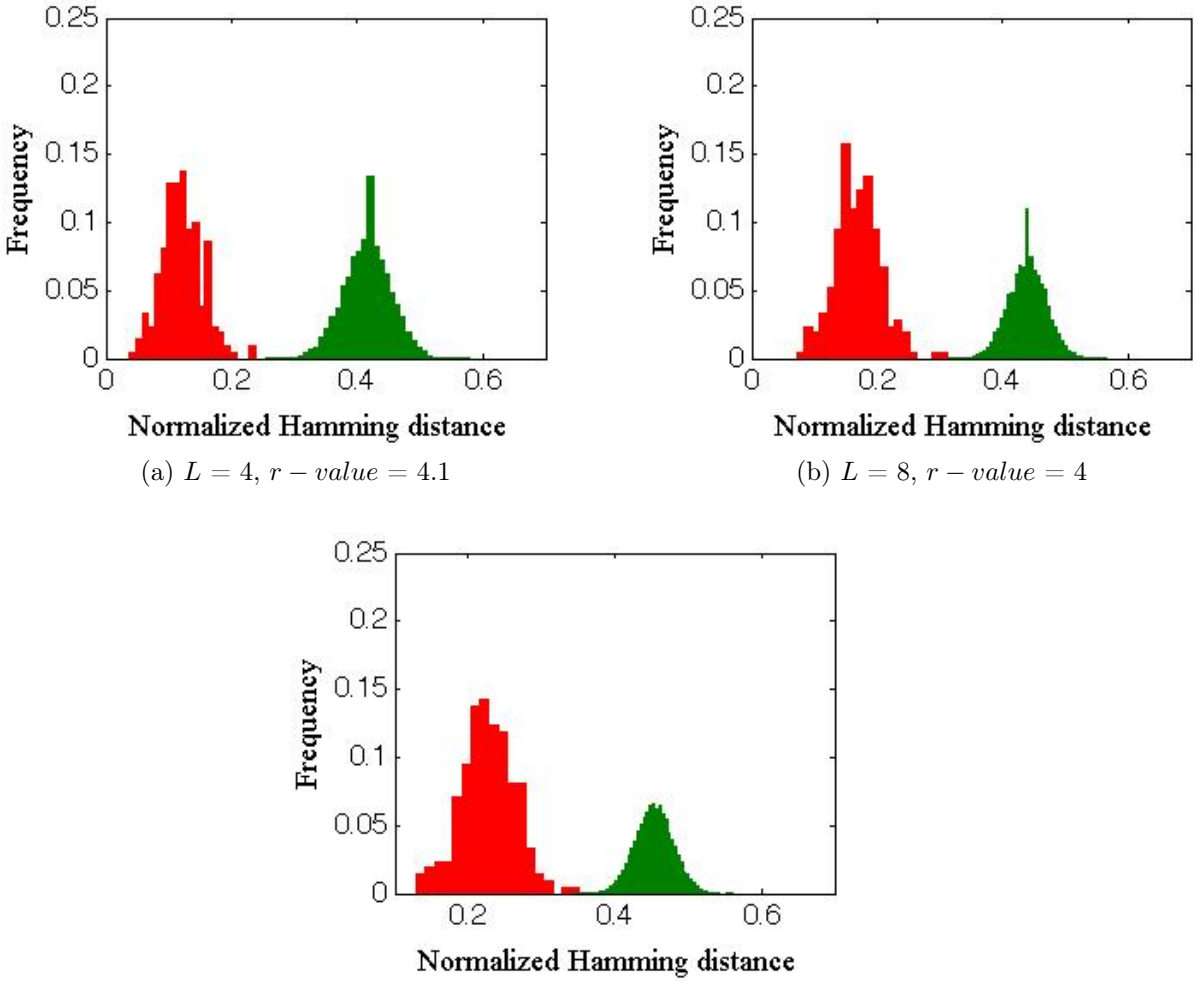


(m) gaussian noise with $\sigma = 0.02$,$r - value = 4.3$



(n) gaussian noise with $\sigma = 0.05$,$r - value = 3.8$

Figure 4.25: Authentic and impostors distributions for several attacks for 120PC and L = 4.



(a) $L = 4$, $r - value = 4.1$

(b) $L = 8$, $r - value = 4$

(c) $L = 16$, $r - value = 3.6$

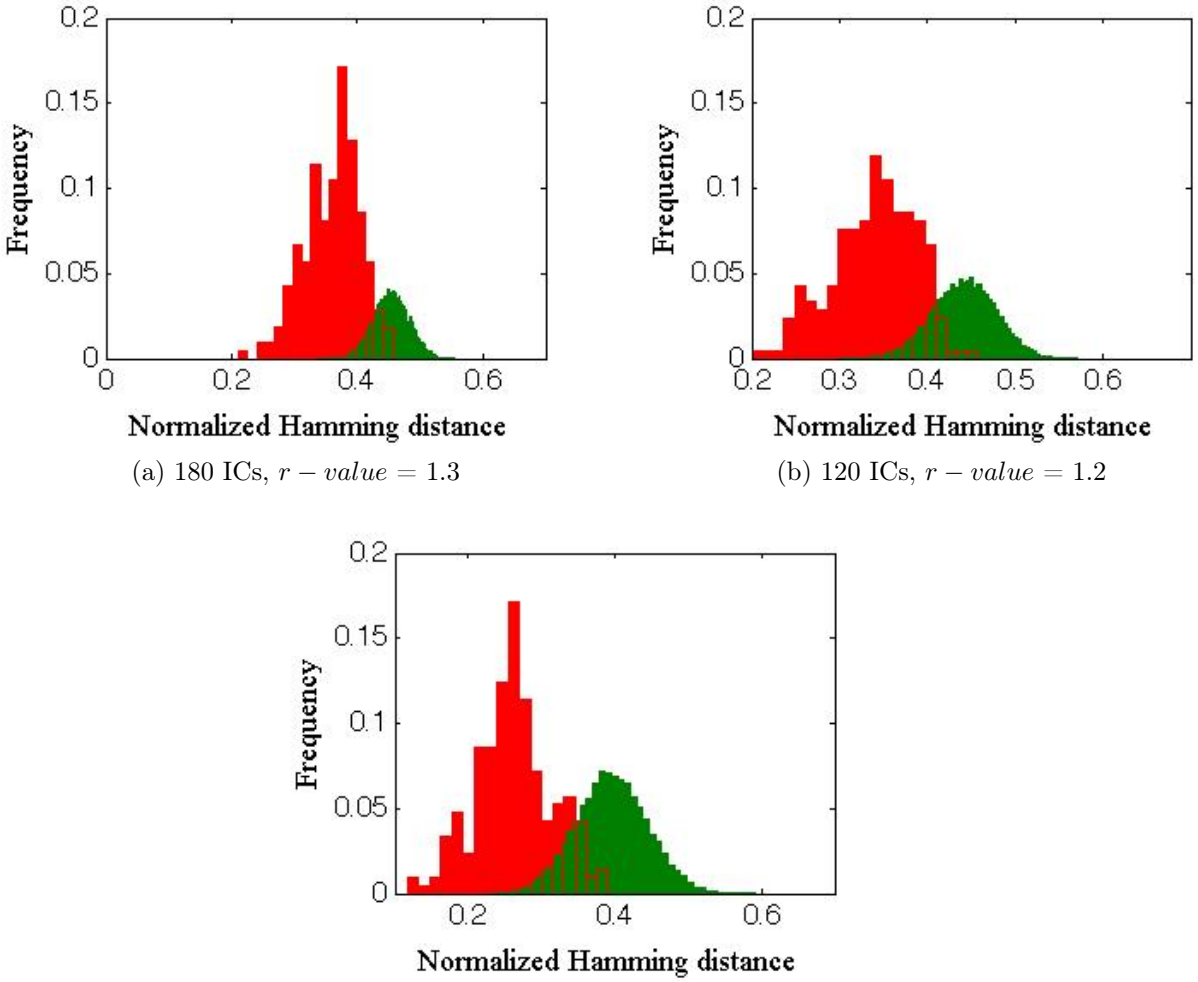
Figure 4.26: Genuine and impostors histograms in the case of JPEG compression with Q = 15, 120 ICs and different quantization levels L.

4.6 Conclusion

This chapter develops a scheme for extracting the intermediate hashes from ID images via ICA decomposition. Two ICA–based approaches for extracting the feature vectors are presented. Several subspace selection algorithms which enforce the robustness of the method are proposed.

The two approaches are tested on simulated data in order to demonstrate their advantages. Almost all images were successfully recognized in both approaches. The performance i.e. *r–value*, varied when using different numbers of ICs and quantization levels. The more ICs are used, the higher the performance is. The same tendency was observed for ICA architecture I when increasing the number of quantization levels. In ICA architecture II the opposite tendency was noticed.

In ICA architecture I the most suitable subspace selection proved to be the local



(a) 180 ICs, $r - value = 1.3$

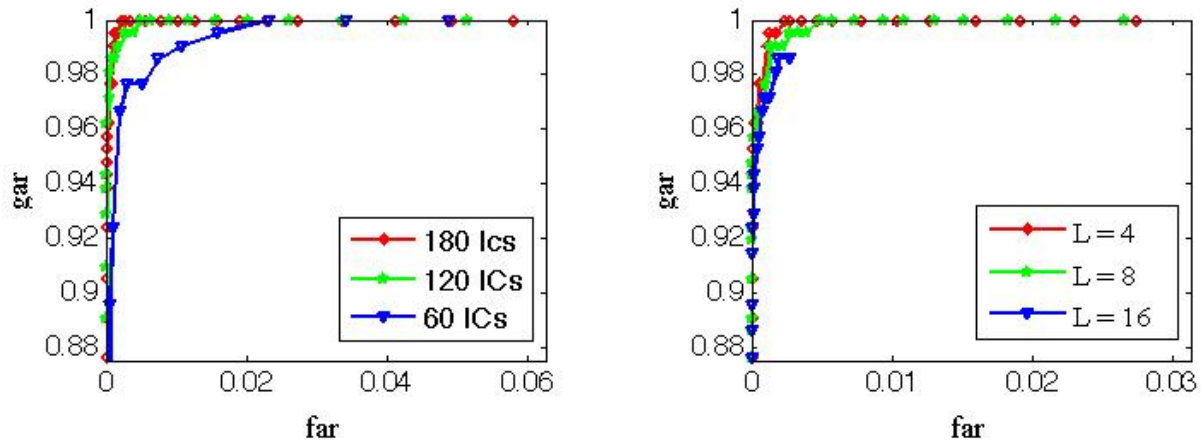
(b) 120 ICs, $r - value = 1.2$

(c) 60 ICs, $r - value = 1.2$

Figure 4.27: Genuine and impostors distribution in the case of AFFINE_2 attack for different number of ICs and $L = 4$.

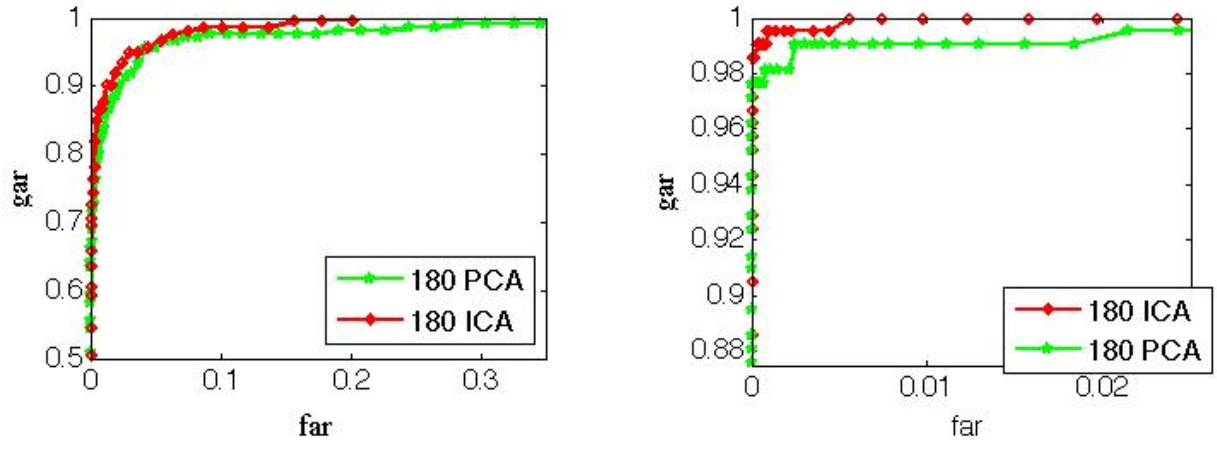
entropic criterion. When comparing the two architectures, ICA architecture I performed better for the cases where full recognition is not achieved, i.e. when the genuine and impostors histograms overlap.

The performances obtained by using the $r - value$ are validated also by the ROC curves and by looking at the genuine and impostors distributions.



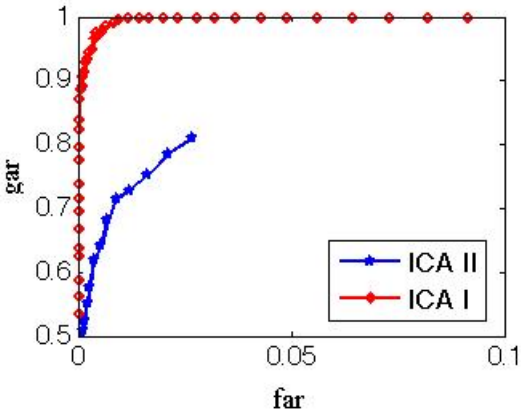
(a) $L = 4$ (b) 180 ICs

Figure 4.28: ROC curves for median filter with a 5 x 5 window. a) $L = 4$ and different number of ICs; b) 180 ICs and different quantization levels.

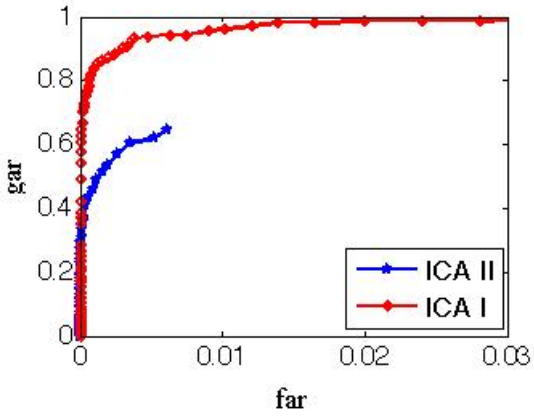


(a) 180 ICs (b) 120 ICs

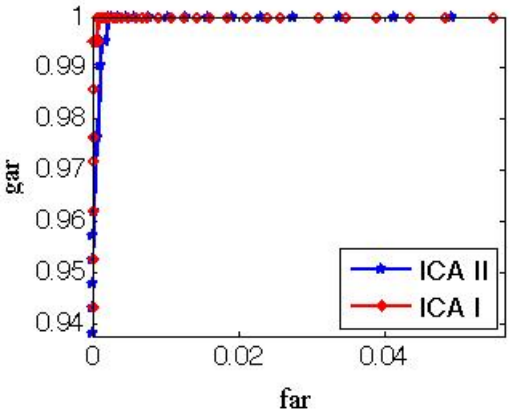
Figure 4.29: ROC curves for PCA and ICAII in the case of: a) median filter with a 7 x 7 window; b) gaussian white noise with $\sigma = 0.05$



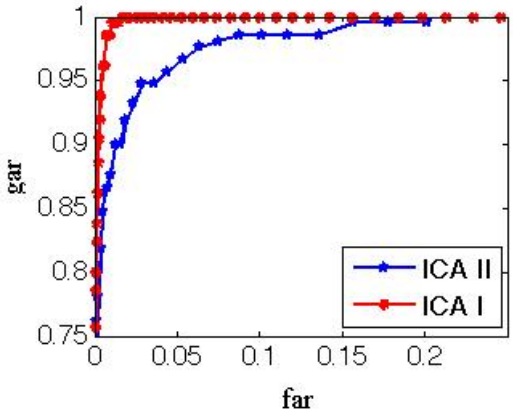
(a) Affine_2



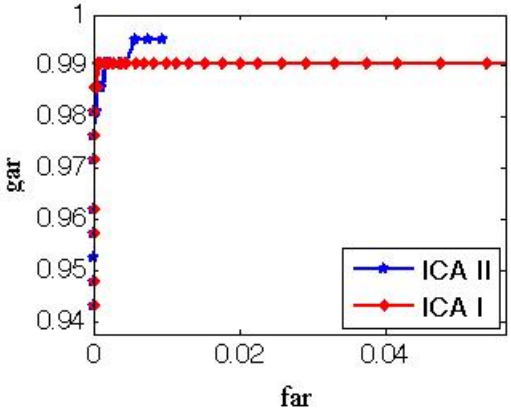
(b) Affine_4



(c) Median 5 x 5



(d) Median 7 x 7



(e) Noise $\sigma = 0.05$

Figure 4.30: ROC curves for ICA I and ICA II under several attacks.

Chapter 5

Print–and–Scan Channel

5.1 Introduction

This chapter presents an unifying framework for the print–and–scan channel. The goal of the chapter is to highlight the fundamental noises and to show the proposed models for these noises and also the solutions developed to cancel or to reduce them.

The devices employed in the process and the way they work are described in section 5.2. Section 5.3 defines the different kind of noise present in the print–and–scan process. The exiting print–and–scan models are then described in section 5.4. Finally, section 5.5 summarizes the ideas discussed in this chapter.

5.2 Print–and–Scan Process

Nowadays the print–and–scan process is used for digital image reproduction and distribution. During the print–and–scan the image is modified. Even if the scanned image may look like the original one, distortions – invisible or visible to the naked eye – intervene in the process. These distortions appear both in the printing and the scanning process, therefore the two processes will be described in the next two subsections. [Lin 1999a]

5.2.1 Printing

A *printer* is a peripheral that communicates with another device, i.e. computer, telephone, automated teller machine (ATM), in order to produce text or graphic output on physical print media, usually the paper.

Over the years, numerous printing technologies have been developed. In modern printer devices the main technologies are the following:

- *toner–based printers*: laser printers are devices capable to create high quality text and graphics on physical media. The hard copy is produced by the direct scanning of a laser beam across the printer's photoreceptor.

- *liquid inkjet printers*: inkjet printers are the most frequent type of printing device used by consumers. In this technology the digital image is reproduced by spurting out droplets of ink onto the paper.

- *solid ink printers* utilize solid ink sticks in place of fluid ink or toner powder. The waxy sticks are melted and scattered onto a spinning drum that fixes the ink on a piece of paper.

- *dye–sublimation printers*: a dye–sublimation printer is a printing device which utilizes the heating process in order to transfer dye to plastic cards, paper or canvas.

- *inkless printers*:

  - *thermal printers*: the printer creates a hardcopy of an image by selectively heating areas of thermal paper. The surface turns black in the regions where it is heated, creating thus the image.

  - *UV printers*: the printer employes a special UV light in order to write and erase the sheet of paper.

Laser printing devices, in comparison with inkjet printers, have a better printing resolution, are faster, with a lower cost per sheet of paper and they do not need special paper as thermal printers.

In our experiments we have used a laser printer. Next, I will briefly describe the principle of laser printers.

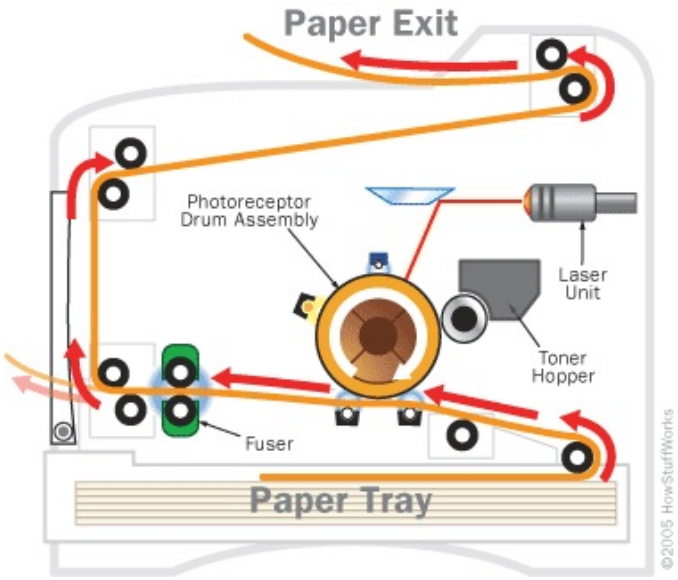


Figure 5.1: The circuit of a sheet of paper in laser printing [www.howstuffworks.com].

The laser printing process consists in the following steps (Fig. 5.1):

a) *raster image processing*: The generation of the image to be printed is done by a Raster Image Processor (RIP), usually built into the printing device. In this stage the image is converted into halftone image by using a page description language such as Adobe PostScript, HP Printer Command Language or Microsoft XML Page Specification.

b) *charging*: The main principle used in a laser printer is static electricity. The charged corona wire (or charged roller) gives a total positive charge to the photoreceptor, a revolving photosensitive drum or belt, able of keeping the electrostatic charge on its coat.

c) *exposing*: While the drum rotates, the laser directs the laser beam through the surface to discharge certain parts. The laser beam reverses the charge on the black areas of the image, leaving a static electric negative image on the photoreceptor – an electrostatic image.

d) *developing*: The printing device covers the drum with positively charged toner (black powder). The fine particles stick to the electrostatic image, onto the areas touched by the laser beam, negatively charged.

e) *transferring*: The image is transferred by pressing or revolving the drum belt onto the sheet of paper. The paper is previously negatively charged by the corona wire. In order to pull the toner powder, the paper charge must be stronger than the one of the electrostatic image on the photoreceptor drum.

f) *fusing*: While the paper passes through the fuser (a couple of heated rollers), the toner powder melts down and merges with the paper fibers. Then the fuser rolls out the paper to the output tray (Fig. 5.2).

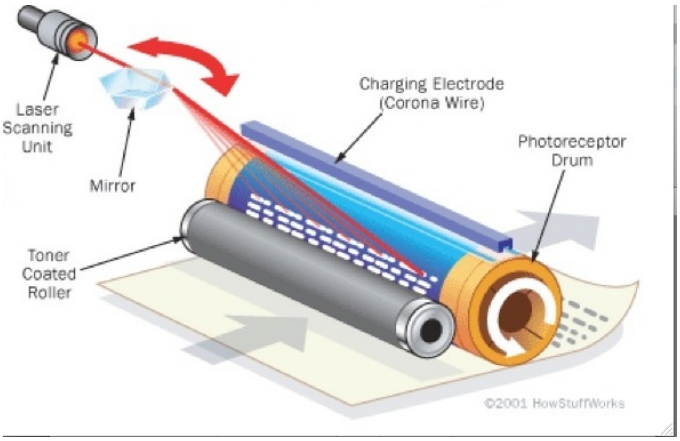


Figure 5.2: The printing process in laser printer [www.howstuffworks.com].

g) *cleaning*: Once the paper is printed, an electrically neutral soft plastic blade removes any surplus of toner from the photoreceptor, placing and depositing it into a dispose

reservoir. A discharge lamp cleans the remaining charge from the drum. Sometimes, when unanticipated events occurs such as paper jam, toner powder may remain on the photoreceptor.

5.2.2 Scanning

A *scanner* is a stand–alone peripheral that captures a hard copy input (image, a document etc.) and converts it to a *digital image*, allowing users to save the data on a computer and reuse it. The scanner technology is widely used. The scanners may be classified as follows:

a) *flatbed scanners*: also known as desktop scanners, are the most commonly employed devices. Their basic principle applies to almost all the other scanners.

b) *sheet–fed scanners* are similar to flatbed scanners with the difference that the document is moved and the scanning head is immobile. The shape of the devices is the one of a small portable printer.

c) *handheld scanners* have the same technology as a flatbed scanner, except that the user is the one moving the scanner. It is not suitable for obtaining high image quality.

d) *drum scanners* are employed in the publishing industry. In order to represent fine detailed image they use the photomultiplier tube as scanning technology. The original document is mounted on an acrylic cylinder and the sensor placed at the center splits the light returned from the document into three colored beams: red, green and blue light.

In our experiments a flatbed scanner (Fig. 5.3) was used. The anatomy of such a device consists in the following parts: charged–couple device (CCD) array or contact image sensor (CIS), mirrors, scan head, glass plate, lamp, lens, cover, filters, stepper motor, stabilizer bar, belt, control circuitry, analog–to–digital converter (ADC).

The main component of a flat bed scanner is the image sensor – CCD array [Taylor 1998] or CIS. The CCD or the CIS receives reflected light from the scanned image and the integrated photodiodes convert it into an electrical signal (electrons). When the reflected light falls on the CCD/CIS, it is converted into electrons. The brighter the light is, the more electrons are produced. The image sensor amplifies these electrons in order to give them a value which can be read by the ADC.

First, the paper is set on the glass plate and the cover is closed. The role of the cover is to provide an uniform background used as a reference point for obtaining the dimension of the document. Next, a lamp is employed to illuminate the document. A belt connected to the stepper motor then gradually moves the scan head across the paper. An angled mirror is used to reflect the image of the scan on a second (or third) mirror, depending on the scanning device. The latter mirror reflects the image onto a lens which will focus the image through a filter on the CCD array. Then the CCD employs an ADC to digitize the image. The digital image will be sent to the scanner's own hardware and from there to the host computer.

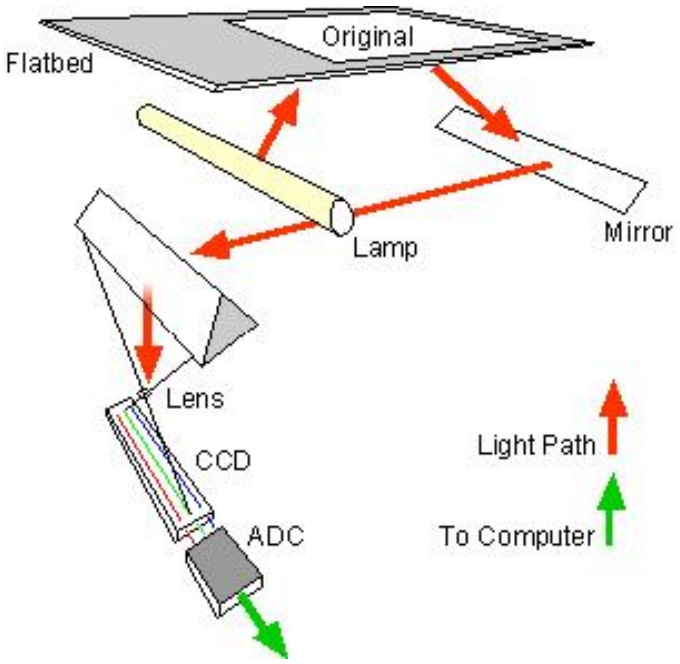


Figure 5.3: The scanning process.

5.3 Noises in Print and Scan Attacks

The print–and–scan process is an elaborate combination of strong and various attacks which introduce modifications on the printed–and–scanned images. When successively printing and scanning a digital image, the output is distinct, even if it appears similar to the naked eye. Fig. 5.4 outlines the different kinds of distortions and noises an image is subjected to when it is printed and scanned. In [Solanki 2006], Solanki *et al.* separated these distortions into three main categories: geometric transformations, non–linear effects and colored noise.

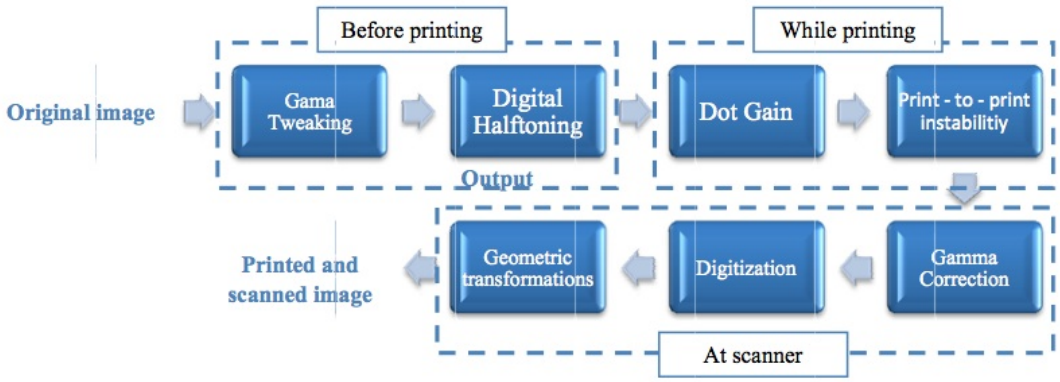


Figure 5.4: Noises in the print–and–scan process.

A short description of distortions is given below:

1) *gamma tweaking* is a non–linear adjustment which takes place in printers. Many printing device vendors modify the transfer characteristics of printers in order to match an uncalibrated monitor. In this way, the printed image appears the same as when displayed on the monitor.

2) *digital halftoning*: before printing, the digital image is transformed into a binary image. This process of converting an image into a binary image is called halftoning. The most common halftoning method represents each image pixel as halftone cell (halftone pattern).

3) *dot gain*: it is a phenomenon causing the printed image to look darker than planned. It is induced by the halftone dots growing in dimension between the original dot and the final printed one. It is a non-linear process mainly caused by colorant spreading around the dot.

4) *print–to–print instability* like banding, which consists in minor variations in the printer's output.

5) *scanner gamma correction*: is necessary in order to compensate the human visual system (HVS). By gamma correction, the grey level of the output pixels is raised at the power $1/\gamma$, where γ is the gamma value of the monitor displaying the scanned image.

6) *digitization*: converts a signal from the analog form to the digital one and introduces quantization errors that can be amplified by the gamma correction. It consists in two stages: discretization and binarization, both occurring simultaneously.

7) *geometric transformations*: while scanning, the image undergoes a series of geometric transformations such as cropping, rotation, scaling and translation. Most of them are man–dependent and even in a careful scanning the effects of geometric transformations cannot be avoided.

5.4 Review of the print–and–scan models

In order to model the print–and–scan process, the following effects should be taken into consideration:

a) Pixel value distortions because of the print–and–scan process;

b) Geometrical distortions – rotation, scaling, translation (RST) –, which may be large–grained (because of paper placement, for example) or fine–grained.

A lot of work on the print–and–scan channel has focused on empirically finding domains that are invariant to some type of distortions. For instance, frequencies in which non–linear effects predominate are ignored or transforms that have some inherent robustness to geometrical distortion are chosen (e.g. log–polar or Radon).

5.4.1 Countermeasures for print–and–scan noises

In [Yu 2005], Yu *et al.* analyze the structure of laser printer and scanner device and propose a print–and–scan model for digital watermarking illustrated in Fig. 5.5.

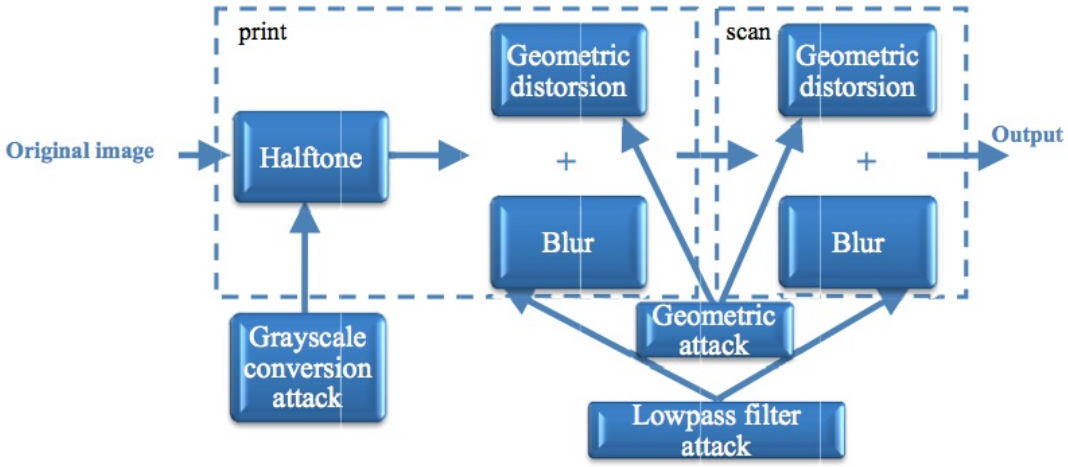


Figure 5.5: Print–and–scan model proposed by Yu *et al.*

The authors do not give a statistical expression of the print–and–scan noise but try to find a countermeasure to the noise. First, they briefly describe the distortion sources in the print–and–scan process.

In the printer device, one source of distortion is considered to be introduced by the optical system when converting the electronic pulse level into an optical one. Another source of distortion is caused by the electrophotographic system which, because of the non–smoothness of the drum and rollers surface, spreads the toner powder non–uniformly on the paper. These two distortions can yield dot gain. The halftoning attack has the most powerful effect on the image. The authors conclude that a countermeasure against halftoning is mandatory.

In the scanner device, the optical system introduces a Gaussian blur, which is similar to adding a Gaussian low–pass filter attack to the image. Another random Gaussian noise is introduced by the stepped motion jitter of the carriage in the scanner. Thermal and dark noise current are introduced by the CCD. The additive and multiplicative noises that appear in the scanning process are difficult to correct but can be minimized by using high resolution scanners.

The geometric distortions are random and specific to the user. Yu *et al.* consider that the geometric distortions are fatal for watermarking methods and propose a countermeasure against this type of attack. It is a three step solution consisting in: pre–processing, detection and correction process.

1) *pre–processing step*: it is known that ID cards have a rectangle frame in order to place

the photograph of the card holder in the right position for good detection. Thus the frame is important to locate the watermark and to identify the geometric distortions. The authors propose frame adding to the original image, frame which will be detected on the printed and scanned image. Next, median filtering and Laplacian sharpening are performed for geometric distortion detection and watermark location.

2) *detection step*: the contour and frame of the pre–processed image are obtained by using a morphological operator. Next, the difference image is computed and used for the inner frame extraction. The output image, defined as the sum of the inner and pre–processed image frame, is superposed with the preset rectangular frame in order to calculate the rotation angle. The scaling factor is determined from the length and distance of the output image contours.

3) *correction step*: it consists in two parts. The first is a coordinate transform in which a conversion matrix is employed to undo rotation and the second is the elimination of interpolation points.

Yu et colleagues divide the countermeasure against grayscale conversion attack in two types:

1) *without blur* which models the grayscale conversion. The scanned halftone image is divided into blocks of M x N according to the dimension of the halftone matrix. Each block has one of the (M x N+1) halftone cells associated. The relationship between x, the original image pixel, and the corresponding number of the halftone cell, y, is

$$x = 256 - (y - 1) * [256/(M \cdot N)] \tag{5.1}$$

2) *with blur*: The blurred halftone image is divided into blocks of M x N according to the dimension of the halftone matrix. Each block is compared with the (M x N+1) halftone cells and the least vector distance criterion is employed in order to associate the block to the corresponding halftone cell. Each original pixel x is then computed as in 5.1.

In [Solanki 2005], Solanki *et al.* model the print–and–scan process as a three components process: mild cropping, coloured noise and non–linear effects. For the coloured noise, two sources of distortions are considered: digital halftoning and the print–to–print instability. Both distortions introduce high–frequency noise in the original image. The non–linear transformations (such as gamma correction) affect the high–frequency, but also the mid–frequency coefficients. The only one that affects all the frequency bands – low, mid and high – is mild cropping. Unfortunately, the purpose of the authors in this article is to achieve data hiding in the low–frequency coefficients, and thus, no countermeasure for coloured and non–linear noise is proposed. The solution to mild cropping is a simple hiding strategy. The only countermeasure proposed is a new method: to approximate and undo the rotation. The method relies on the knowledge of the printer's

halftoning algorithm. When scanning at high–resolution, the halftone cell can be captured and employed to estimate the rotation angle. The estimation is done by using the fact that halftone cells are orientated with a certain angle with the horizontal.

In [Zhang 2009], both geometric distortions and pixel values distortion introduced by the printing and scanning process are considered. The difference value D between the pixel of the original image I and the distorted one (that suffered geometric distortions), I', is defined by:

$$D(i,j) = I'(i,j) - I(i,j) \tag{5.2}$$

After geometric transformation, the pixel *(i,j)* is shifted to the position *(i+*Δi*, j +* Δj*)*. Δi and Δj correspond to the geometric distortions magnitude.

The authors consider gamma tweaking (γ_p) and gamma correction (γ_s) as non–linear phenomena affecting the pixels value. After gamma tweaking, the light brightness B can be written as

$$B = I(i,j)^{\gamma_p} \tag{5.3}$$

While scanning, gamma correction is performed on B. Thus

$$I'(i,j) = I(i,j)^{\frac{\gamma_p}{\gamma_s}} \tag{5.4}$$

Zhang at al. add random noise N to their model and obtain the following final expression for$D$:

$$D_{i,j} = I'(i,j) - I(i,j) = I(i-\Delta i, j-\Delta j)^{\frac{\gamma_p}{\gamma_s}} + N - I(i,j) \tag{5.5}$$

While $\gamma_p > \gamma_s$ the geometric distortions are amplified and D is distributed in the entire image. The solution to this problem proposed in the paper is to add down–sampling operation to obtain low–resolution. First the original image is down–sampled by k:

$$I_w(i,j) = \frac{\sum_{x,y=1}^{k} I[(i-1)*k+x, (j-1)*k+y]}{k^2} \tag{5.6}$$

The same operation is applied on the scanned image I'. By down–sampling, the effect on the image of the non–linear geometric distortions is reduced in the ratio by k^2.

5.4.2 Statistical Models of the Print–and–Scan Chain

Few attempts have been made to develop statistical models of the print–and–scan channel. In fact the main references in this regard are [Amiri 2009, Degara-Quintela 2003, Lin 1999b, Malvido 2006, Villán 2005, Voloshynovskiy 2004, Kundu 2006]. The most detailed of these is [Malvido 2006], which takes into account digital cameras as well as scanners on the image acquisition side. Lin and Chang [Lin 1999b] propose a model considering both pixel value distortion and geometric distortion. [Villán 2005] presents a simpler model with fewer parameters to be estimated. Similar with [Amiri 2009, Degara-Quintela 2003], the channel is modeled as a non–linear transformation and additive, image–dependent noise. [Kundu 2006] propose a tuning mechanism to measure print–and–scan transformation which can correlate the image before printing and the one obtained after scanning.

5.4.2.1 Lin and Chang Model

Lin and Chang [Lin 1999b] outline the properties of the print–and–scan process and propose a model. The authors consider that the distortions introduced in the print–and–scan process can be divided into two categories:

1) pixel value distortions caused by luminance fluctuations, contrast modifications, gamma correction, chrominance variations, and blurring. These are distortions perceptible to HVS;

2) geometric distortions consisting in rotation, scaling, cropping (RSC). These kinds of distortions may not always be perceptible to the HVS, but introduce modifications in the image and thus cause problems in image authentication;

Lin and Chang distinguish RSC from RST used in pattern recognition, as by using image processing Graphical User Interface (GUI)'s it is possible to scan only a portion of arbitrary size from the original image.

They propose a model of the pixel value distortions under the print–and–scan attack focussing on luminance, as this is where information embedding is to take place.

Pixel value distortion Model

Assume there is a virtual, continuous finite support image, I, which is reconstructed from the original image I_0:

$$I(x,y) = \begin{cases} \sum\sum I_0[n_1,n_2]\delta(x-n_1X_0, y-n_2Y_0), & x \in [-\frac{X_1}{2}, \frac{X_1}{2}], y \in [-\frac{Y_1}{2}, \frac{Y_1}{2}] \\ 0 \quad \text{otherwise} \end{cases} \tag{5.7}$$

where X_0 and Y_0 are the inverse of dots per inch values in directions x and y, and X_1 and Y_1 are the bounds of the image support. The printed image is a dithered version of I, with additional noise. The scanned image I' is modeled as

$$I'(x,y) = K[I(x,y) * \tau_1(x,y) + (I(x,y) * \tau_2(x,y)) \cdot N_1] \cdot s(x,y) \tag{5.8}$$

where K is the responsivity of the detector defined in 5.9, s is the sampling function, N_1 is a white gaussian noise with a higher power near the edges, τ_1 is the scanner's optical point spread function, and τ_2 respectively a high–pass filter employed to enhance the higher noise variance near the edges.

$$K(x) = \alpha(x-\beta_x)^\gamma + \beta_K + N_2(x) \tag{5.9}$$

where N_2 represents the thermal and dark current noise. The other terms of the equation represent the combined AC, DC and gamma adjustments in the printer and scanner [Lin 1999b].

Geometric distortions Model

In the print–and–scan process, images are discretized at both ends of the process while in the intermediate stages they are continuous [Lin 1999b]. For this reason, Lin *et al.* propose a continuous–domain definition of geometric distortions. The distorted image I_G is represented as:

$$I_G = GI \tag{5.10}$$

where G is the geometric distortion operator (single or multiple RSC).

5.4.2.2 Villan Model

In [Villán 2005] a simpler model of the channel is proposed:

$$Y = \phi(X) + Z(X) \tag{5.11}$$

where X is the gray value of the input image, Y is the gray value of the output image, $\phi(.)$ is a non-linear function and Z is a noise modeled by a GGD with parameters depending on X. In practice, the model is fitted by putting each gray–scale value through the channel multiple times and estimating $\phi(X)$ as the conditional mean $\mu_{Y|X}$.

(a)

Figure 5.6

The empirical estimate of $\mu_{Y|X}$ is different for each combination of printer and scanner. For some combinations, the mapping is sigmoidal, with a variance $\sigma_{Y|X}$ that is bell–shaped. By approximating Φ with the appropriately chosen functions and by using a

(b)

(c)

Figure 5.6: Lena image: a) original image; b) after the model of [Villán 2005]; c) after print–and–scan process.

shape parameter of 1.5 in the GGD, we obtain the result shown in Fig. 5.6 on the Lena image.

5.4.2.3 Degara–Quintela Model

In the context of 2D bar codes, a print–and–scan model is proposed in [Degara-Quintela 2003] and enhanced in [Malvido 2006] [Villán 2005]. The basic model takes the following form (Fig. 5.7). Let $u(x, y)$ be the pixel value and $v(x, y)$ be the distorted pixel value.

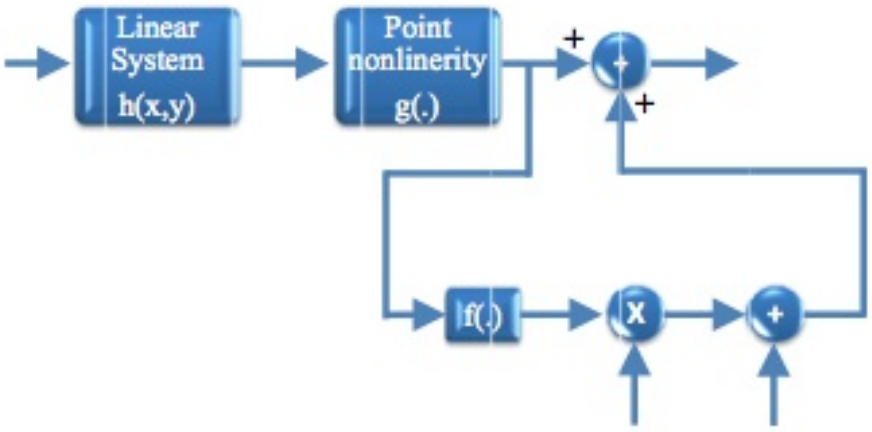


Figure 5.7: Print–and–scan model proposed in [Degara-Quintela 2003].

Then

$$v(x, y) = g(w(x, y)) + \eta(x, y) \tag{5.12}$$
$$w(x, y) = u(x, y) * h(x, y) \tag{5.13}$$
$$\eta(x, y) = f(g(w(x, y)))\eta_1(x, y) + \eta_2(x, y) \tag{5.14}$$

The linear filter $h(\cdot)$accounts for the blurring caused by the low–pass filtering in the printer (η_p) and scanner (η_s)

$$h(x, y) = \tau_p(x, y) * \tau_s(x, y) \tag{5.15}$$

The functions $f(\cdot)$ and $g(\cdot)$ are nonlinear. The response of the image detector, $g(\cdot)$, is:

$$g(w) = \alpha w^{\beta} \tag{5.16}$$

where α and β are device–dependent parameters.

The term $\eta(\cdot)$ in 3.11. is an additive noise composed by an image–dependent random component and by image–independent random component (3.13.). $\eta_1(x, y)$ and $\eta_2(x, y)$ are the CCD noise (Poisson noise) and thermal noise, which may be modeled as Gaussian white noise.

5.4.2.4 Amiri and Jamzad Model

The model in [Amiri 2009] empirically estimates noise as image–dependent and image–independent noise of the print–and–scan channel . Similarly to [Villán 2005], in order to estimate the image dependent noise, the effect of print–and–scan on different gray levels and for different γ values is analyzed. They set the mean and variance of the image–dependent noise to depend on the gray level and the gamma correction parameter and find that the image–dependent noise can be modeled by a Gaussian distribution with parameters that depend on the gray level value. Similarly to [Degara-Quintela 2003], they fit the parameters with polynomials.

For the image–independent noise, Amiri and Jamzad print and scan N times different constant gray level images by defining the difference between two print and scan versions of the same image as noise. They model the image–independent noise by empirical fitting of a logistic distribution to the observed noise.

Finally, to account for the influence of neighbouring pixels on the pixel value (because of dot gain, for instance), a classifier is designed to distinguish classes of the images based on their complexity. For each class a neural network is trained in order to obtain the image resulting from neighbour influence. The network's input is provided by the pixel values of the neighbours and the output is the deduced modified pixel value.

5.4.2.5 Kundu Model

Kundu *et al.* propose in [Kundu 2006] a tuning mechanism to model the print–and–scan process (Fig. 5.8). The model is described by a simple equation:

$$Y_i = T(X_i) \tag{5.17}$$

where X_i, Y_i is the pixel gray value before printing, respectively the pixel gray value after scanning and i is the gray value ($i = 0, ..., 255$).

T_1 and T_2 from Fig. 5.8 are transformations that have taken place when passing from halftoning to printing and printing to scanning. The total transformation function T is defined as the convolution of T_1 and T_2.

The authors estimate the function T by experimenting on a synthetic image with different strips of known gray levels (from 0 to 255). The image is halftoned by using a halftoning technique similar to the one used in [Yu 2005] and then printed. The printed image is scanned in order to produce the corresponding image output. The gray level values of the scanned image are different from those of the original image because of T_1 and T_2 transforms. Kundu and colleagues estimated the function T by using polynomial interpolation:

$$Y_i = \sum_{j=0}^{r} a_j X_i^j, \ i = 0, ..., 255 \tag{5.18}$$

where a_j are the polynomial coefficients.

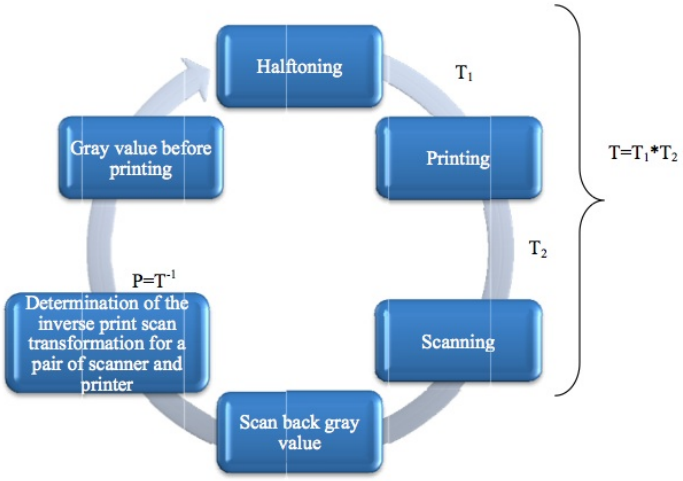


Figure 5.8: Print–and–scan model proposed by Kundu *et al.*

5.5 Conclusion

This chapter presents the complex process of printing–and–scanning. The printing and scanning devices were described and divided into main categories. The laser printers, in comparison with ink jet printers are faster, have a better printing resolution, with a lower cost per sheet and do not need special paper like the thermal ones. The most common image scanners, flat bed scanners, have high scanning resolution, are inexpensive and faster. Nowadays they are employed in most graphic–arts operations. The main distortions present in the print–and–scan process were also described.

Mathematical expressions that model the print–and–scan process were introduced. Two important observations can be made: 1) most of the work in this domain has focused on finding different spaces that are robust to print–and–scan noise; 2) few statistical print–and–scan models were developed by researchers.

Chapter 6

ICA based hashing and Print–and–Scan Channel

6.1 Introduction

This chapter analyzes the effects of the scanner noise on the printed image. First the experimental print–scan chain is presented in section 6.2. Next, two types of noise, halftoning and scan noise, present in the process are described in section 6.3. In subsection 6.3.3 a model for scan noise is illustrated. The different types of noises present in the scan process are experimentally identified. The experimental results presented in section 6.4 show robustness to the print–scan process. The genuine and impostors distributions are quite well separated. The system performance is tested for the two architectures described in chapter 4 and for different ICA algorithms presented in chapter 4.

6.2 Print and Scan Chain

The print–scan chain used for the experiments is presented in Fig. 6.1. The physical transformation from the digital image to the printed image is performed by printing it with a HP LaserJet P3005dn printer [Haas 2009]. Once the image is printed on the document, the rightful owner can make use of it to prove his identity and access secured areas. In the case of loss or theft of the document, unscrupulous individuals can take advantage and modify the content of the image or make unauthorized copies. The fraudulent copies can be then rescanned and reprinted at a high quality resolution on a new document. At a certain point in time, the original image, the unauthorized copy of the image or the modified image, is scanned when passing the document through a verification system, using a specific scanner device – here a HP ScanJet 3600. Thus, the image can be recovered in order to analyze it and to identify whether it is authentic or not (in this way the authenticity of each document containing an image can be verified). It would be helpful to specifically predict the effects of the scan process in order to decide if the image is authentic or is a malicious copy.

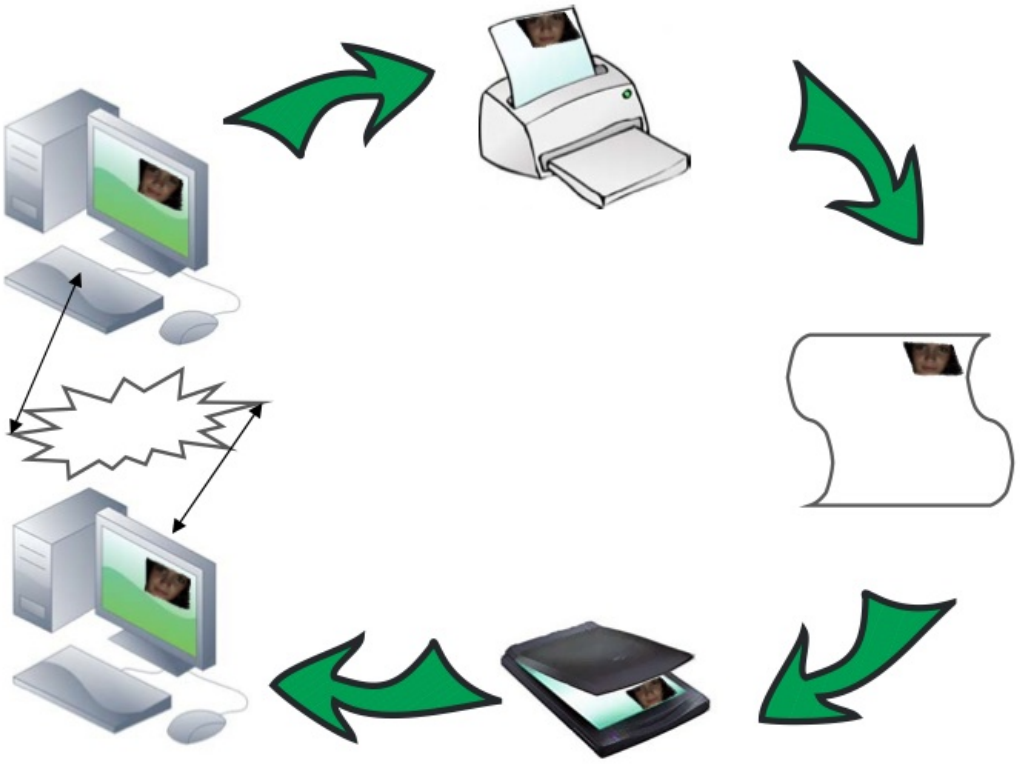


Figure 6.1: The print–and–scan chain..

6.3 Noises

As seen in chapter 5, in the print–scan process many factors contribute to image distortion. In this section I will focus my attention only on haltoning and scan noise.

6.3.1 Haltoning

Halftoning is an important source of noise in the print–and–scan process. It plays a significant role in the image reproduction process due to the limited number of colors in the printing device. The digital grayscale images use a palette of 256 different shadows of gray, while the black and white printing devices are restricted to only one color, i.e. black, and should somehow represent the 256 different gray levels only by placing black dots on a white paper. By the process of printing or not printing black marks, the original continuos image is transformed into a binary image (a bitmap, containing only 1's and 0's– the value 1 stands for printing a black dot, while 0 for leaving the space empty). This conversion from a continuous–tone image to a bitmap representation is called *Halftoning* or *Screening* [Vikas 2005]. Since the human eye acts like a low–pass filter, the printed and not–printed black marks are blurred, creating thus the impression of continuous–tones of gray (Fig. 6.2). Depending on the printed dots distribution, different degrees

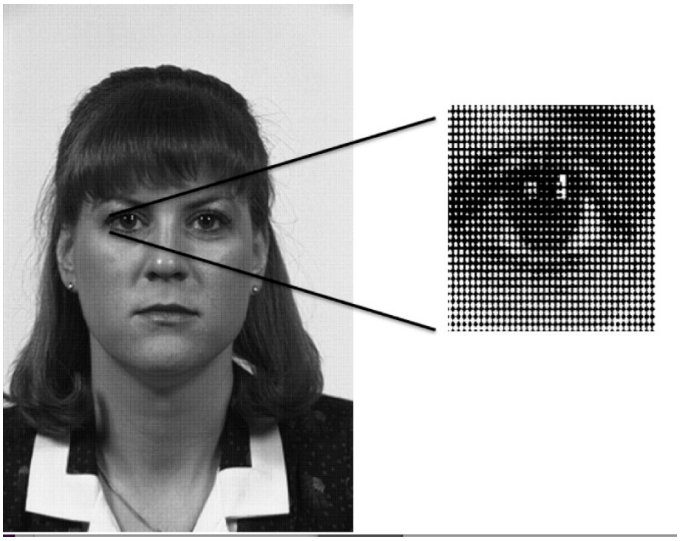


Figure 6.2: Gray–scale ID photograph reproduced as a halftone.

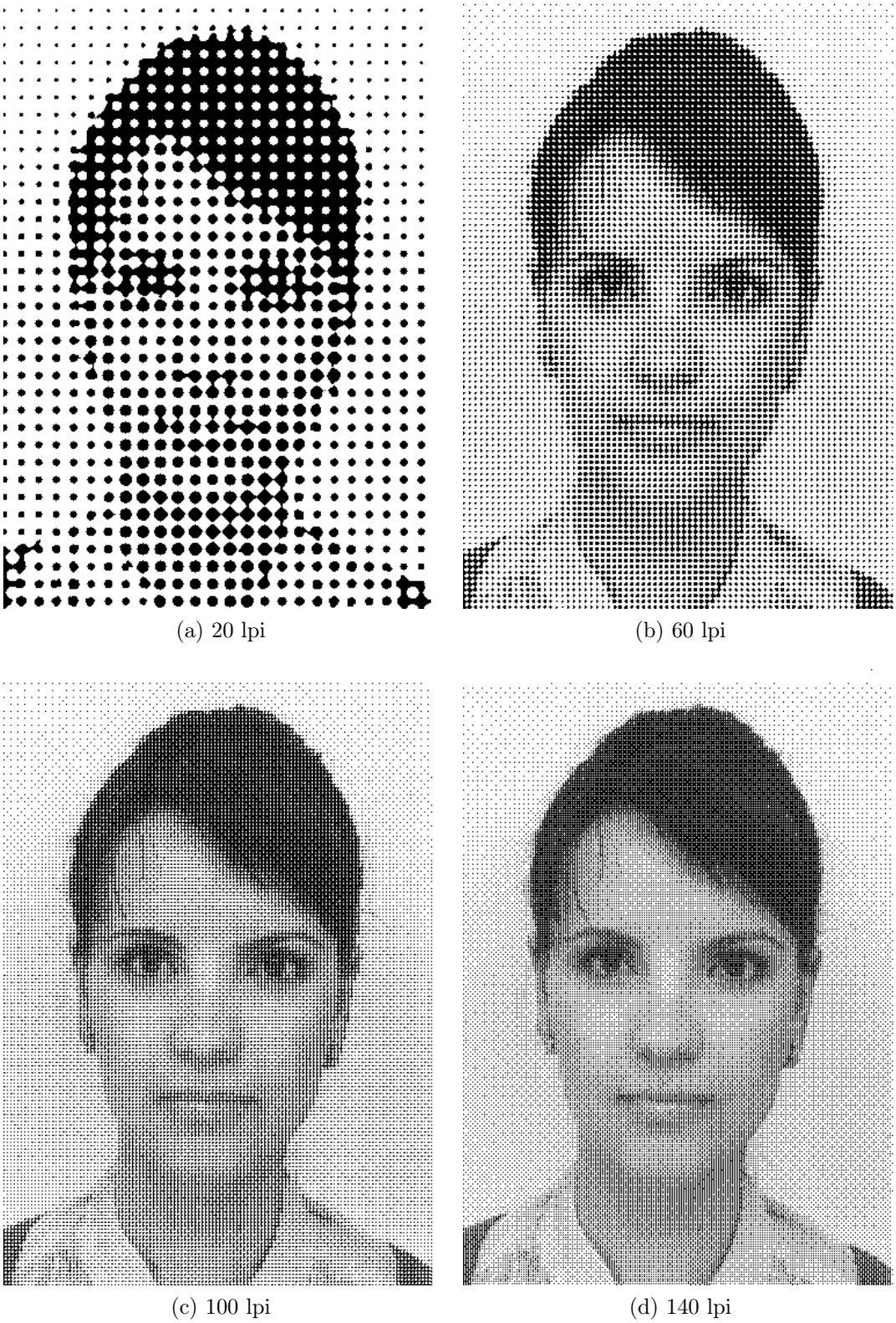
of image fidelity can be achieved. Randomly arranged and isolated dots lead to high quality images according to the human visual system [Lau 2008] by preserving fine details and sharp edges. An important issue is that not all the printing devices are capable to reproduce isolated dots; they introduce printing artifacts which contribute to degrade the details that the dot distribution should usually preserve. For this reason, many printing devices make patterns of cluster dots. The purpose is to find the propitious distribution of dots for a specific device and to generate the patterns efficiently in order to minimize the visibility of the artifacts.

The most natural way of halftoning is to represent image parts by a *halftone screen (halftone cell)*. Each such halftone cell contains a number of smaller dots known as *microdots*. In Fig. 6.6, two 8x8 halftone cells representing two different gray level tones are shown. The halftone cell on the left side illustrates a gray tone of 4/64 (2x2 black microdots from a total of 64 microdots), while the halftone cell on the right a 25/64. For an 8x8 halftone cell a total number of 65 gray tones can be obtained by using the relation given in 6.1.

Let us recall that the number of microdots in an area of a square inch is defined as dots per inch (DPI)– considered the printing resolution – and the number of halftone cells in a square inch of area is defined as Lines Per Inch (LPI) – being considered the screen frequency. As it can be seen in Fig. 6.3, for a constant number of *dpi* the quality of the image differs when tuning the number of *lpi*.

$$number\ of\ gray\ tones = \frac{dpi}{lpi} + 1 \tag{6.1}$$

Beside *lpi*, other factors like dot shape and angle can be modified in order to obtain a better quality of the printed image. In Fig. 6.4 dot clusters vary in both size and shape according to the gray tone [Lau 2008] while in Fig. 6.5 it is in size and angle. The most common dot shapes are round, square, and elliptical [Coudray 1996] and



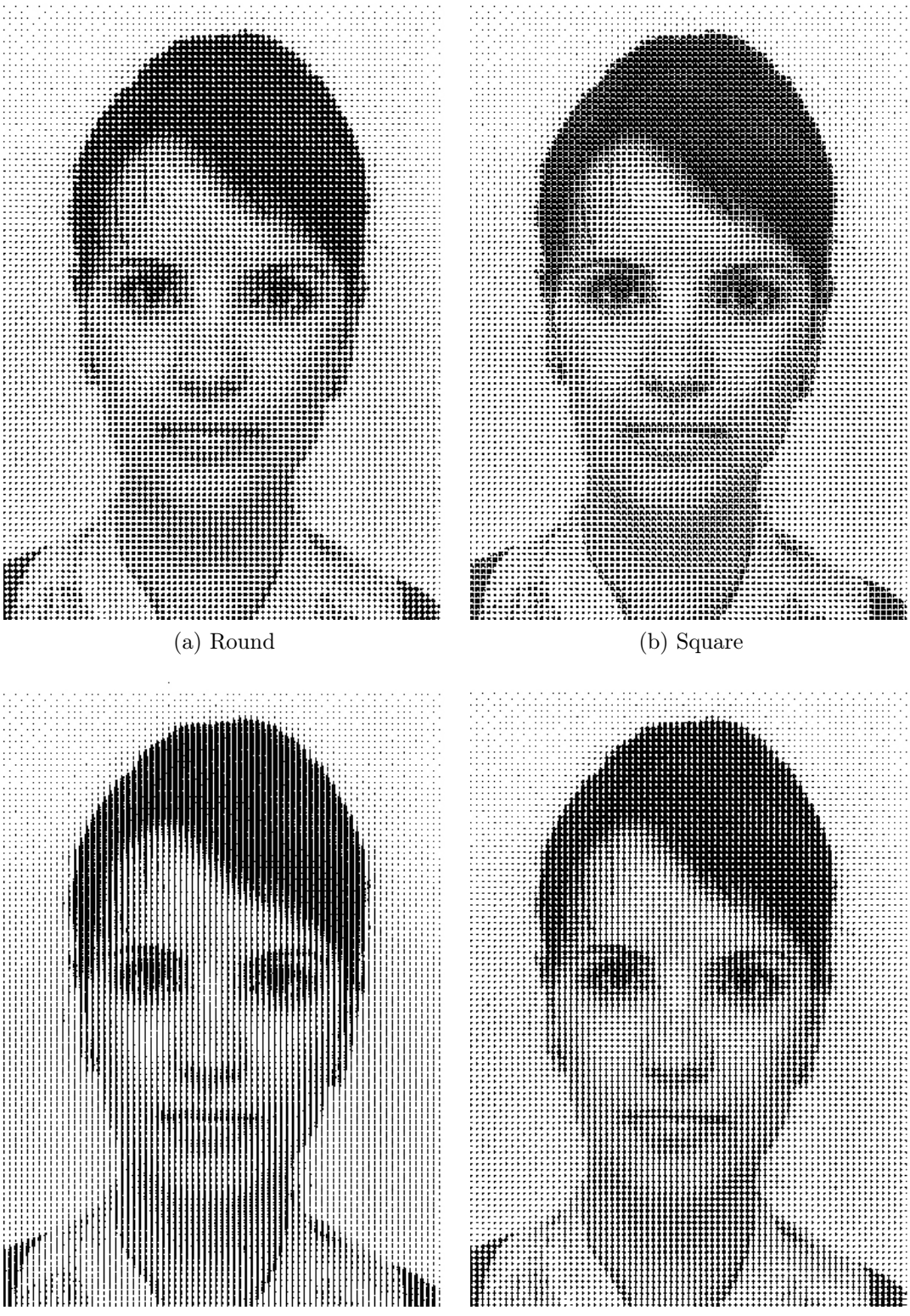
(a) 20 lpi

(b) 60 lpi

(c) 100 lpi

(d) 140 lpi

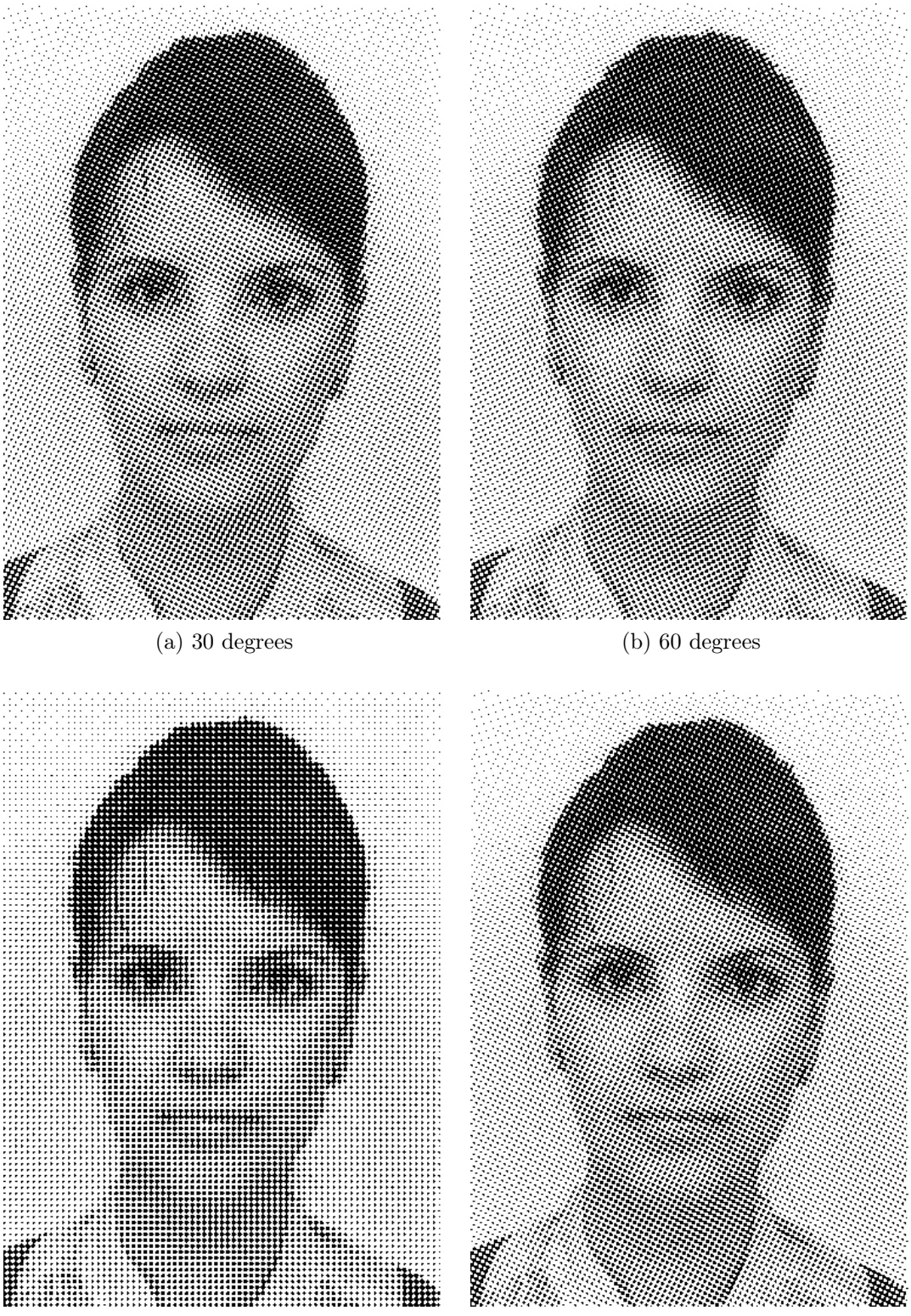
Figure 6.3: Halftoned images for a constant print resolution of 300 dpi screen frequency.



(a) Round (b) Square

(c) Line (d) Diamond

Figure 6.4: Different dot shapes for halftoning.



(a) 30 degrees

(b) 60 degrees

(c) 90 degrees

(d) 120 degrees

Figure 6.5: Different angles for the dots.

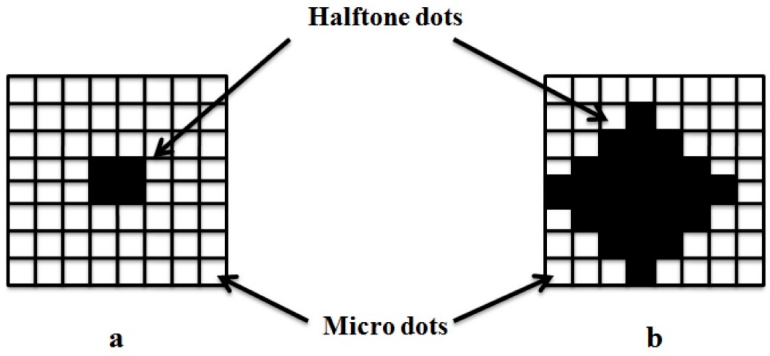


Figure 6.6: Two halftone cells: a) 4/64 gray level; b) 25/64 gray level.

directional artifacts are less noticeable for the human visual system at an angle of 45 degrees [Campbell 1996].

The different halftoning methods that exist in the literature can be mainly classified into three categories: amplitude modulation methods (AM), frequency modulation methods (FM), and AM – FM hybrid methods.

6.3.1.1 AM Halftoning

AM halftoning methods represent different shades of gray by a regular pattern of clustered dots with different size. The dark gray shades are represent by large clusters while the light gray shades by small ones as shown in Fig. 6.7. These type of techniques present the advantage of low computation, good print stability and resistance to artifacts [He 2004] but they are not robust to moiré effect.

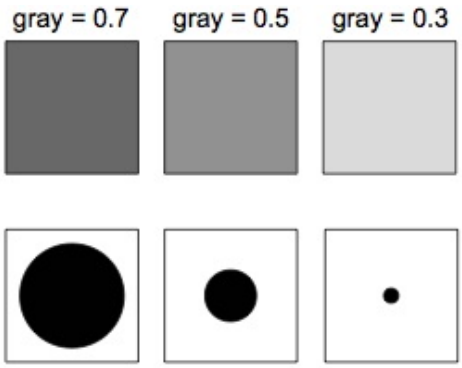


Figure 6.7: Halftoned images for a constant print resolution of 300 dpi screen frequency.

Some of the most used AM methods are described below.

a) *table halftoning* consists in replacing small parts of the original image by their associated halftone cell [Ekdemir 2011]. Let us assume that each 2x2 pixel area is to be replaced by a 8x8 halftone cell. Since the size of the halftone cell is 8x8, only 65

different gray shades can be obtained. The mean gray value of the 2x2 pixel region is replaced by the corresponding gray tone candidate as shown in Fig. 6.8.

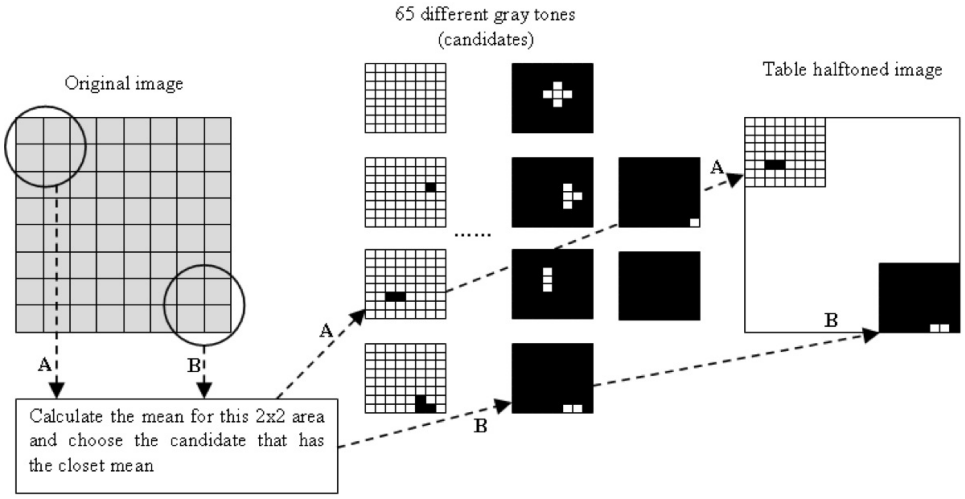


Figure 6.8: Table halftoning example. Each 2x2 submatrix in the original image is replaced by a 8x8 halftone cell. Since the table size is 8x8, 65 different gray levels can be represented.

b) *threshold halftoning* consists in using a threshold matrix t which is tiled over the image. Each pixel value at position *(i,j)* is compared with the same position value in matrix t as summarized in equation 6.2. The pixel value from the original O is replaced in the halftoned image H with a 0 (empty dot), respectively an 1 (black dot), if the value is greater, respectively smaller, than the threshold.

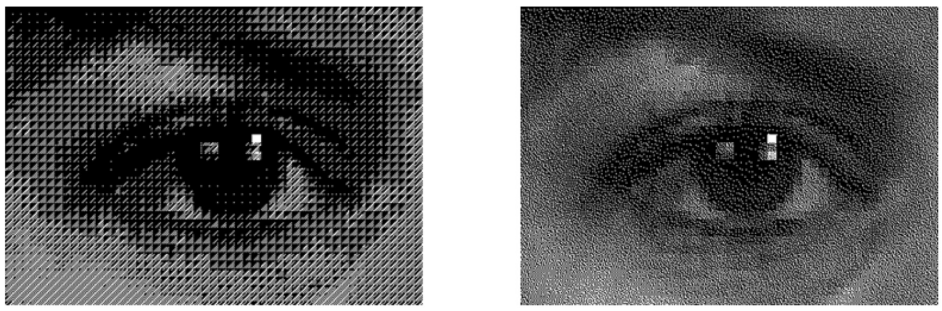
$$H(i,j) = \begin{cases} 1 & \text{if } I(i,j) \geq t(i,j) \\ 0 & \text{otherwise} \end{cases} \tag{6.2}$$

c) *ordered dithering* is a significant threshold technique. The ordered dither methods can be divided into two groups: pattern dithering and diffusion dithering.

In pattern dithering the consecutive thresholds are located in spatial proximity [Ekdemir 2011] producing in the halftoned image clustered dots in the center of the halftone cells (Fig. 6.9 a)). In this approach a trade–off between the number of gray levels and the resolution is required. In the diffusion dithering technique, the consecutive thresholds are dispersed leading to dispersed dots in the halftone cells (see Fig. 6.9 b)). In comparison with the other method, here there is no need to take into account the trade–off between the number of gray levels and the resolution.

6.3.1.2 FM Halftoning

In FM halftoning the different shades of gray are represented by modulating the frequency and keeping the dot size invariant. These methods typically have higher spatial resolution and resistance to moiré effects [He 2004] than the previous ones.



a) Pattern Dither **b) Diffusion Dither**

Figure 6.9: Halftoned image by a) pattern dither and b) diffusion dither. Improved detail rendition is obtained by applying the latter technique.

A well known FM technique is error–diffusion. Each of the original image pixels are quantized to either 1 or 0 by a neighborhood operation (Fig. 6.10). The value of the output pixel is not dependent only on the input pixel, but on a neighborhood of the input pixel. In order to decide whether a pixel from the original image I has to be printed or not, the pixel is compared with the print threshold T. If the value is higher or smaller than the threshold a 1 or a 0 is placed in the halftoned image O in order to suggest if the pixel has to be printed or not. Since the halftoned image has binary values and the original image continuous ones, an error is detected. The error e is computed as the difference between the desired output at that position and the printed level. The error is then distributed to the pixels that have to be printed as decided by the error filter.

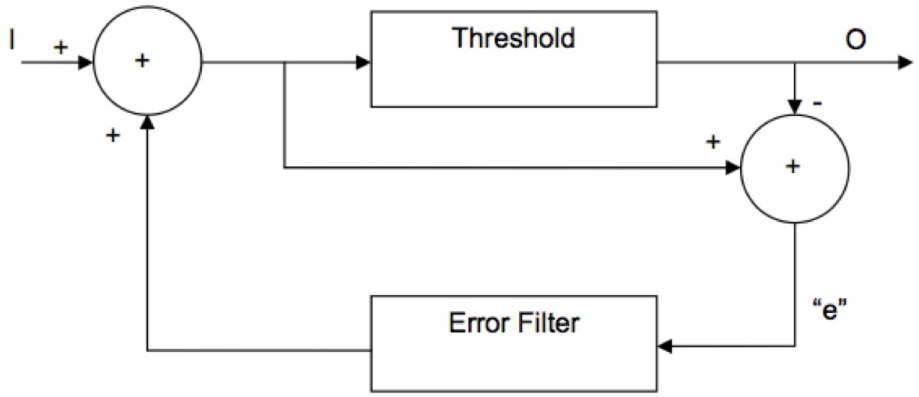


Figure 6.10: Error–diffusion halftoning scheme.

6.3.1.3 AM–FM Hybrids

Nowadays, some printers achieve resolutions larger than 1 200 dpi, the limits of FM halftoning have also been reached, just as AM's was in 1990 before the introduction of

low–cost color. Researchers, therefore, begin to look at AM–FM hybrids producing dot clusters that vary, according to tone, in both their size and spacing [Aoki 1998, Lau 1998]. When considering the reproduction of monochrome images, AM-FM hybrids are, in general, capable of producing patterns with lower visibility (higher spatial resolution) compared to AM and, if stochastic, do so without a periodic structure adding an artificial texture to the printed image. With some amount of clustering, these halftones are easier to print reliably and with little variation in the resulting tone.

6.3.2 Scanning Noise

In the scanning process of converting a printed image into a digital version of that image, various factors have influence on the acquisition process. In the scanning procedure, each image to be scanned is placed on the scanner's flatbed and if it is not placed correctly, the image can suffer a mild rotation. The rotation angle is usually smaller than 3 degrees since the corner of the image to be scanned can be aligned with the corner of the flatbed with a proper accuracy [Solanki 2005].

Another important process that occurs during scanning is *gamma correction*. Scanned images are stored into a computer and displayed on a monitor. It is known that every computer monitor has an intensity to voltage response curve, which is a power function with parameter γ meaning that if a message is sent to a computer monitor specifying that a certain pixel should have an intensity equal to x, the monitor will display a pixel with an intensity equal to x^{γ}. The most common gamma value is 2.2, which means that the intensity value displayed will be less than the one we wanted. In order to have a correctly displayed scanned image on a monitor, the image data generated at the scanner is "gamma corrected" (i.e. raised to a power $1/\gamma$).

These factors are deterministic and can either be controlled by the user (like scanner resolution) or corrected (like geometrical distortions and gamma correction). A significant damage on the scanned image is caused by random factors. In the case of our application, it is helpful to know the noise statistics in order to correctly design the feature vectors and to improve the authentication performance.

When digitizing a wedged document with a flatbed scanner, as here, the principal sources of noise are the CCD and the carriage motor. Several types of noises are depicted at CCD level [Baier , Kodak 2005]:

1. The shot noise, also known as statistical or Poisson noise, defines the fluctuation of the number of photons detected by the CCD cells, during the exposure time, that varies according to a Poisson distribution. The shot noise is an inherent physical limitation of the CCD.

2. The reset noise of the sense capacitor that accumulates the electrons generated inside the CCD cells. This charge is periodically released, in order to be converted into a voltage by a source follower amplifier. At the beginning of each new cycle, the capacitor must be reset at an initial voltage. This voltage has random variations called reset noise. Most CCD manufacturers include circuitry that eliminates this noise.

3. The readout noise of the on-chip output stage (the follower amplifier) consisting of a white noise, generated by the amplifier load resistor, and the flicker noise, also called 1/f noise, generated by the amplifier MOSFET transistor. The flicker noise is frequency dependent, while the white noise is not. At high frequencies, the white noise dominates.

4. The thermal noise, also called dark current noise, has two components: dark current non-uniformity and dark current shot noise. The former, resulted from the fact that each line pixel generates a slightly different amount of dark current, can be eliminated by subtracting a reference from each line. The latter can be only reduced by cooling the CCD.

From the above list, most of the noises can be compensated or reduced but the shot noise remains the most significant source of noise. The shot noise is visible in the white areas of the image, when the scanner exposure-time is short enough to avoid CCD saturation.

The CCD scan line is moved vertically, down the length of the bed, by a stepping motor that is pulsed to move step by step. Since the motor precision is limited, the step length varies randomly around a fixed value. This fluctuation is called jitter. The mechanical components of the scanner have also a reset noise i.e., at each new scan, the CCD platform is reset at an initial position which slightly fluctuates from a scan to another. In the scanned image, both jitter and reset noise generate a noise that is gray level dependent.

6.3.3 Noise Model

For the same printed image, the binary features vector issued from different scans are not identical because of scan noise. In order to put into evidence the effects of scan noise at various stages of features extraction, a series of tests were done at high and low resolution. First, a blank page was scanned at 300 spi. The gray level histogram of blank page image is a mirrored Poisson distribution because of CCD shot noise. The distribution is truncated at 255 because of CCD saturation (Fig. 6.11).

This random variation of the white pixels, ranging between 240 and 254 in the case of our experiment, represents a primary source of noise in any scanned image.

Next, two uniform gray level images – light and dark grays – were printed and then scanned thirty times. At high resolution, the histogram of the scanned gray images exhibits a bimodal distribution: beside the mirrored Poisson noise, still present, a Gaussian like distribution appears around level 80 (Fig. 6.12.a and 6.12.b), very probably because of toner non-uniformity (Fig. 6.12.c and 6.12.d). The modes weights depend on printed dots density i.e., on image gray level.

In the dark gray image (Fig. 6.12.d), the dots are no longer separated because of *dot gain*, a printing defect in which the dots dry at a larger size than specified by the printing plate. It increases with the printing resolution. The dot gain contribution to image histogram is visible in the gap between the two modes. As the dot gain becomes stronger, this gap tends to be filled up.

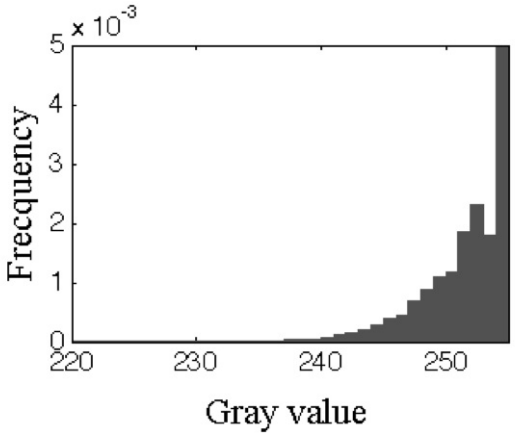


Figure 6.11: Histogram of a blank page image scanned at 300 spi with HP ScanJet 3600 scanner. The white pixels have a mirrored Poisson distribution because of CCD Poisson noise.

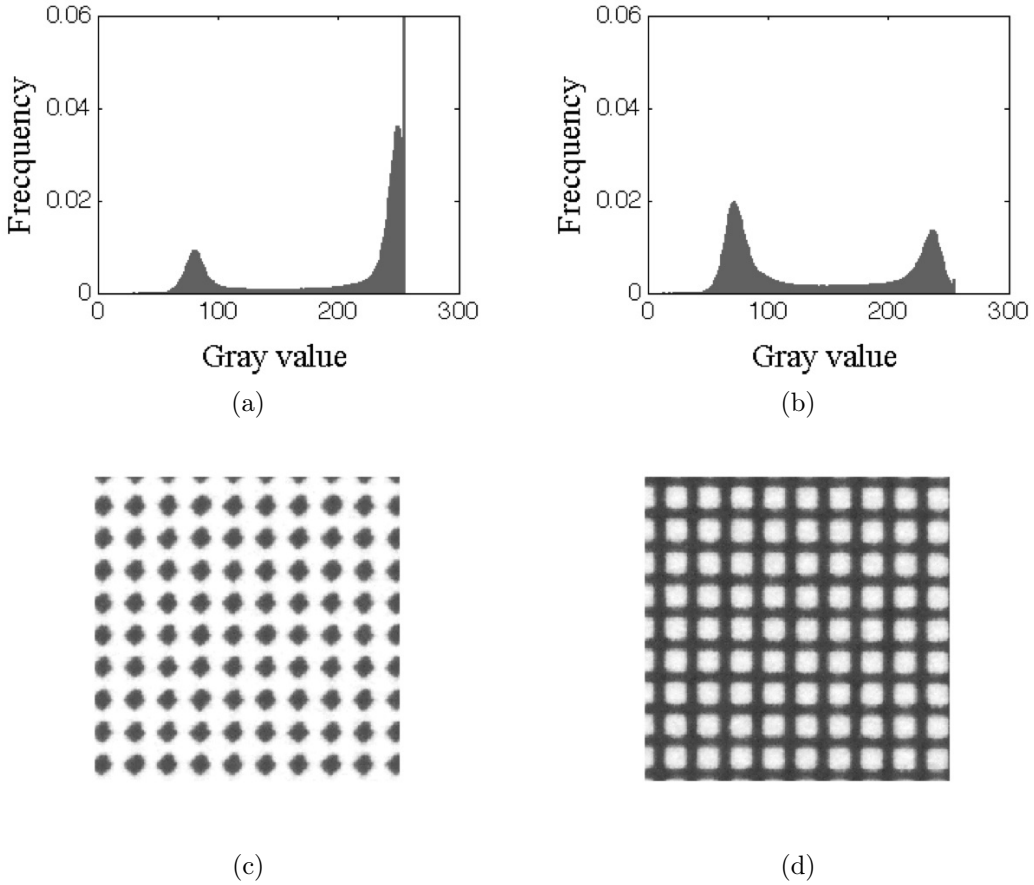
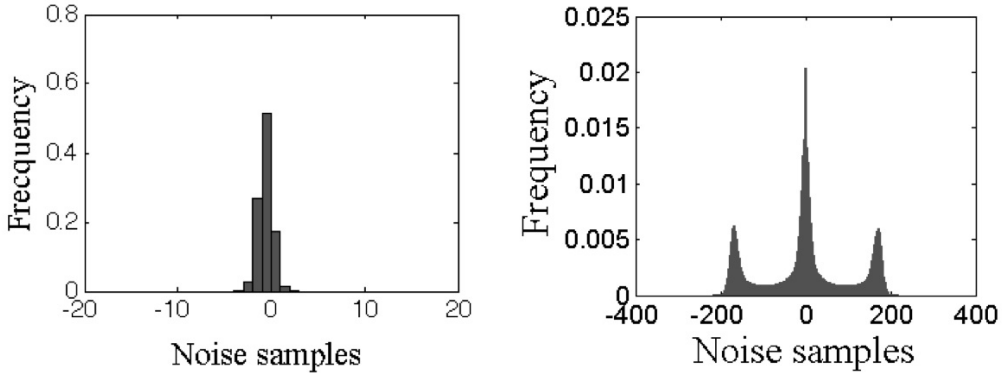


Figure 6.12: Histograms of the two uniform gray images at 300 spi resolution: a) light gray histogram; b) dark gray histogram c,d) upper left corner of light gray and dark gray at 1200 spi, respectively.

Since our algorithm for features extraction is based on a linear analysis of the image, we consider as scan noise the difference between two scans of the same image. With this definition, in the hypothesis of an ideal scanner mechanical components, the scan noise should appear only on the non-printed area and its distribution should be the auto–correlation of the mirrored Poisson distribution (it is known that the distribution of the sum of two independent r.v. is the convolution of their distributions; in the case of subtracting two i.i.d. r.v., the convolution becomes auto–correlation). Indeed, the histogram of two scans difference is symmetrical with respect to zero in conformity with an auto-correlation (Fig. 6.13.a). However, its standard deviation is far from what we expected. Moreover, the standard deviation may vary a lot when other pairs of scans of the same image are subtracted. For the uniform light gray image, the scan variance is in the range of [37.45, 11281], while for the dark gray one it is in the range of [85.15, 16383] (Fig. 6.14a) and 6.14b).



(a) (b)

Figure 6.13: Noise distribution for a light gray image with a) smallest variance, b) highest variance.

This variation is brought by the scanner mechanical component that introduces two types of noise: jitter and reset noise. Because of jitter, the dots in the two scans do not perfectly match (Fig. 6.14c). The result is a supplementary noise, this time in the dot area, with a distribution equal with the auto-correlation of the Gaussian-like distribution. Consequently, the distribution in Fig. 6.13a) is, in fact, the superposition of two modes, both symmetrical with respect zero. Again, the weights of the two modes depend on dots' density, which means that scan noise is dependent on the image gray level.

At the beginning of a new scan, the carriage motor is reset at an initial position. The slight fluctuation of this position produces dots mismatch that may be important (Fig. 6.14d). This mismatch introduces two supplementary modes in the noise distribution, which increases, significantly, the noise standard deviation (Fig. 6.13b). It is the reason why the scan noise variance varies with the couple of scans. Usually, the motor reset noise is not specified, being masked by geometrical distortions of translation type. In our experiment, it is visible since the printed sheet of paper is not removed from the scanner plate for successive scans.

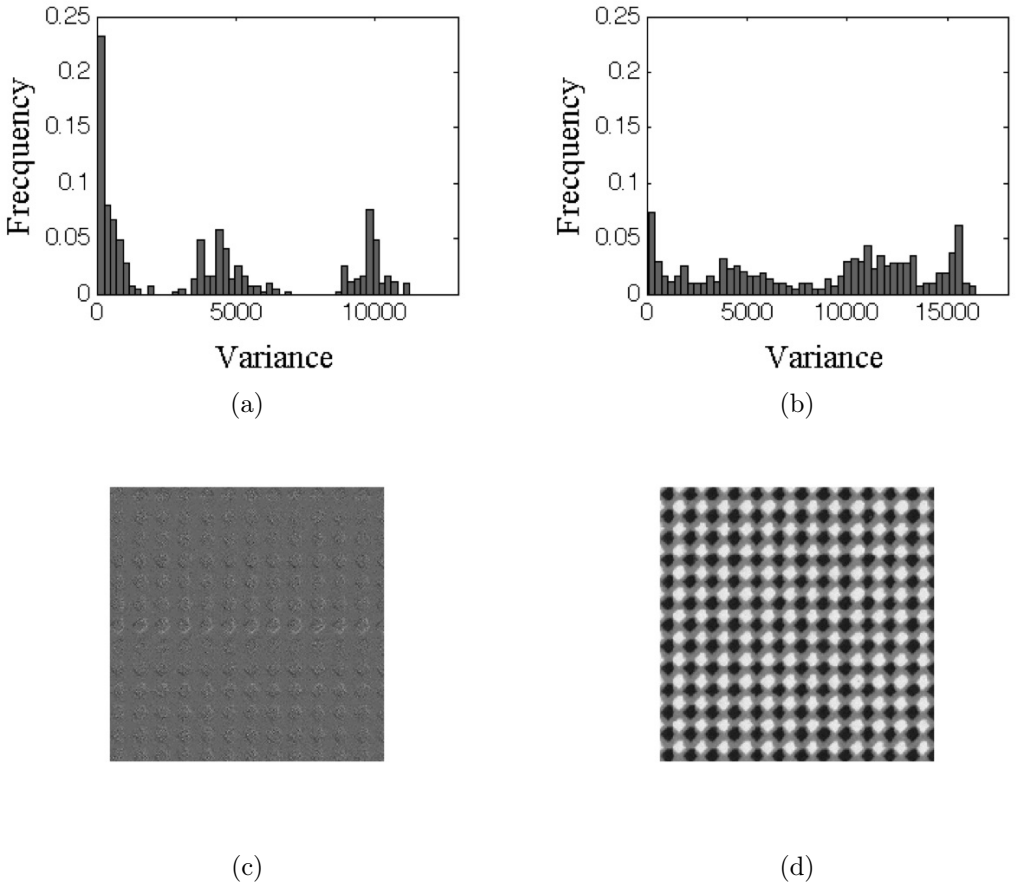


Figure 6.14: Variance distribution for a) a light gray, b) a dark gray, c) difference image of two scans of light gray, f) difference image of two scans of dark gray.

Both jitter and reset noise contribute to scan noise by an amount that is dependent on image gray level. Our experiments with two uniform images showed that scan noise is higher in the dark images (Fig. 6.14.a and 6.14.b).

At a low resolution, after resizing at 384x256 pixels, the histograms of scanned gray images look like in Fig. 6.16.a and 6.16.b. The two modes melt down in a unimodal distribution that depends on image gray level. The scan noise histogram remains symmetrical with respect to zero; its standard deviation varies with the considered pairs of scans (Fig. 6.16.a and 6.16.b).

The print–and–scanned faces have the same behavior. Fig. 6.15.c and 6.15.d show the scan noise histograms for two faces, a light–skin face and a dark–skin one, respectively. The minimum variance case was considered for both images. As in the case of uniform gray images, the noise distributions are symmetrical with respect to the origin, but their variances are different. The noise variance is in the range [0.45, 41.6] for the light–skin face image and in the range of [0.66, 1.13×10^3] for the dark skin face image (Fig. 6.15.e and 6.15.f).

By projecting the image on the basis components, the scan noise is also projected. Fig. 6.17.a shows the histograms of the scan noise on the first IC, in the case of the

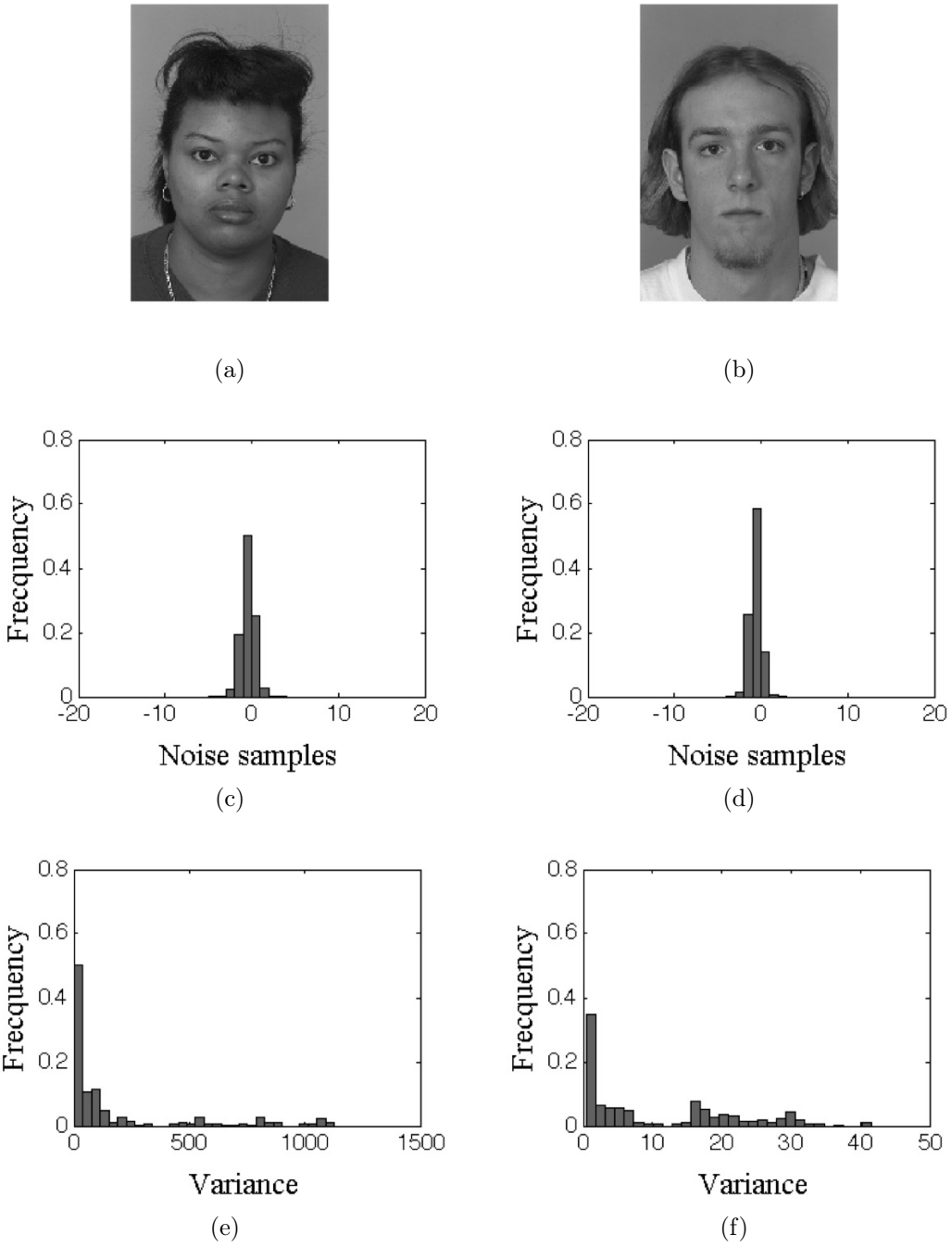


Figure 6.15: Noise samples for the minimal variance and the noise variance of image a) (left side) and image b (right side), respectively.

light-skin face (465 noise samples from the 31 scans). The scan noise variance on the 180 components is in the range of [0.012, 0.25] (Fig. 6.17.b).

It is interesting to note that, by rejecting 60 components (from the 240 obtained after PCA pre–processing) according to an entropic criterion to get the current 180 components, 60% of them having the highest scan noise variance (noise variance between [0.02,

0.16]) are simultaneously rejected. This shows that components with high entropy are the most noisy ones (Fig. 6.18).

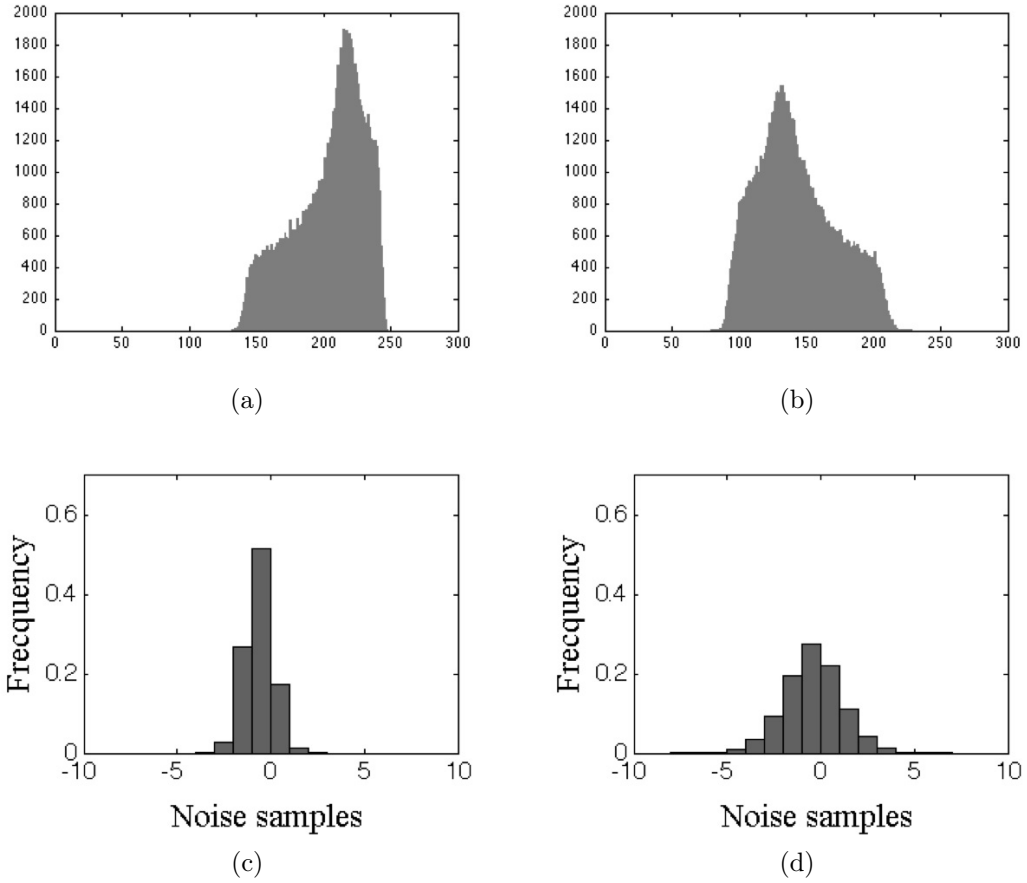
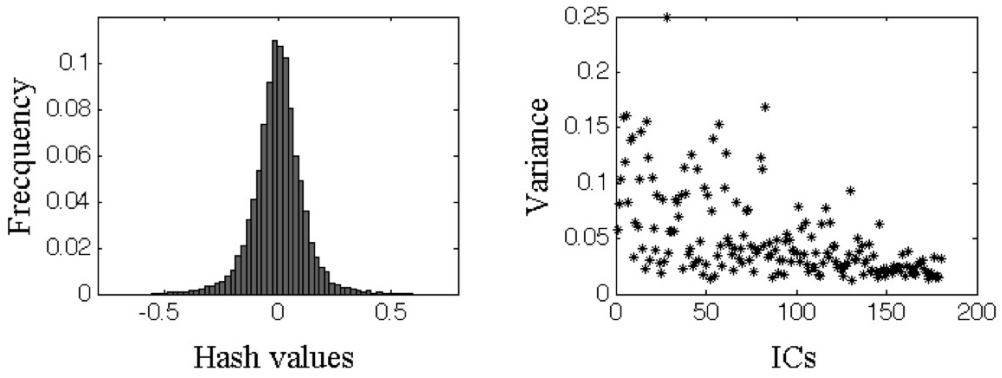


Figure 6.16: Histogram (a) and noise distribution (c) for a uniform, light gray image; histogram (b) and noise distribution (d) for a uniform, dark gray image.

6.4 Results

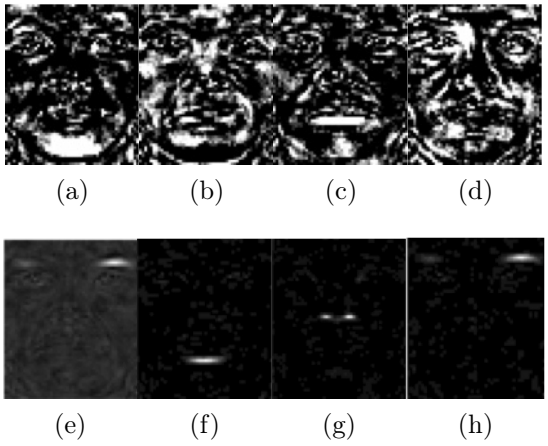
A training set consisting of 300 gray level images of 384x256 pixels was used for learning the basis vectors. The images in the training set were printed as ID photographs, scanned and analyzed according to the methodology described in Section 3. For 300 images a maximum of 300 components can be extracted by using ICA in Architecture I. In order to reduce the length of feature vectors, the least significant principal components were discarded in the preliminary PCA stage of FastICA. Only the first 180 components, carrying about 98.3% of the signal energy, were kept. The corresponding 180 components, derived by FastICA, represent the learned subspace basis.

For the test, other 30 images of FERET database was considered. Each image, once printed, was scanned 31 times. This way, a test set of 930 images was constituted.



(a) (b)

Figure 6.17: a) Distribution of the noise projection obtained for the less noisy components. b) Noise variance for all the 180 components.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 6.18: Four basis components rejected according to the entropic criterion (a – d), having noise variance greater than 0.02. Four selected components (e – h), having noise variance lower than 0.02.

The printing stage of the print/scan experiments was done by using a HP LaserJet P3005dn printer. The images were printed at 600 dpi and 75 lpi, with halftone screen at 90 degrees angle and round shape dots.

A HP ScanJet 3600 scanner was used at low resolution, more precisely 300 spi, like in some other image authentication in the print–and–scan context [Yu 2007]. The additional scans were done without removing the printed sheet of paper from the scanner plate. After scanning, a resizing at 384x256 pixels was done, in order to retrieve the size of the original ID images.

Furthermore, according to the methodology described chapter 4, the scanned images were registered to the same coordinates as those of the training set, cropped and resized once more at 60 x 50 pixels. The eyes and mouth coordinates, used for registration, were determined manually.

By projecting the test images onto the learned basis vectors, a series of 180 coefficients was obtained for each image. These coefficients, quantized on 8 levels and encoded by using a 3 bits Gray code, provide a 540 bits features vector.

6.4.1 Genuine and impostors distributions

In order to evaluate the scan noise influence on the binary feature vectors, we compared them by using Hamming distance. The genuine (13 950 samples) and impostor (202 275 samples from the 930 test images) distributions are drawn in Fig. 6.19. If the 540 bits of the binary features vectors were independent one from each other, the expected Hamming distances distribution for impostors would be a binomial distribution with parameters $p = 0.5$ and $N = 540$ [Daugman 1993]. Since the bits of the feature vectors are not independent, the Hamming distances of the impostors can be modeled by a binomial distribution with $p = 0.3335$ (the mean of the empirical distribution). The standard deviation of the binomial distribution is given by

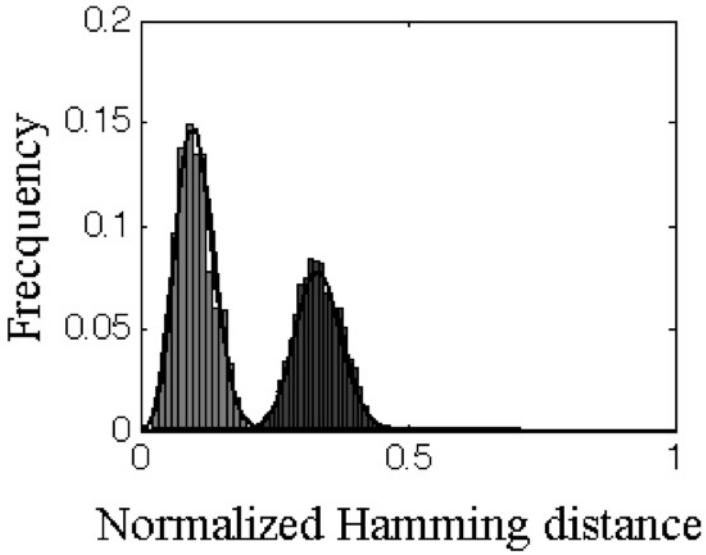


Figure 6.19: Genuine (light gray) and impostors (dark gray) scores for 180 components and L = 8 quantization levels. The binomial distributions used as models are drawn with a continuous black line.

$$\sigma = \sqrt{pq/N} \tag{6.3}$$

where $q = 1 - p$ and N, the number of trials, is equal to 120 (because the standard deviation of the empirical distribution is 0.0431). A theoretical curve of the binomial distribution with such parameters is shown in Fig. 6.19.

Regarding the genuine distribution, it can be observed that the normalized distances between two different scans of the same photograph is never zero because of the noise introduced by the scanner. The mean value of the Hamming distances of different scans of the same ID photograph is $\mu = 0.1023$ and the standard deviation is $\sigma = 0.0344$. The

distribution can be modeled by a binomial distribution with parameters $p = 0.1023$ and $N = 78$, by applying equation (6.3).

6.4.2 ROC Curves

The two distributions shown in Fig. 6.19 are not well separated. An overlap can be observed in the region of 0.18 to 0.22. This overlap gives the FAR and GAR values employed to design the ROC curves and to identify the best threshold d_0.

Table 6.1: Areas under ROC curves

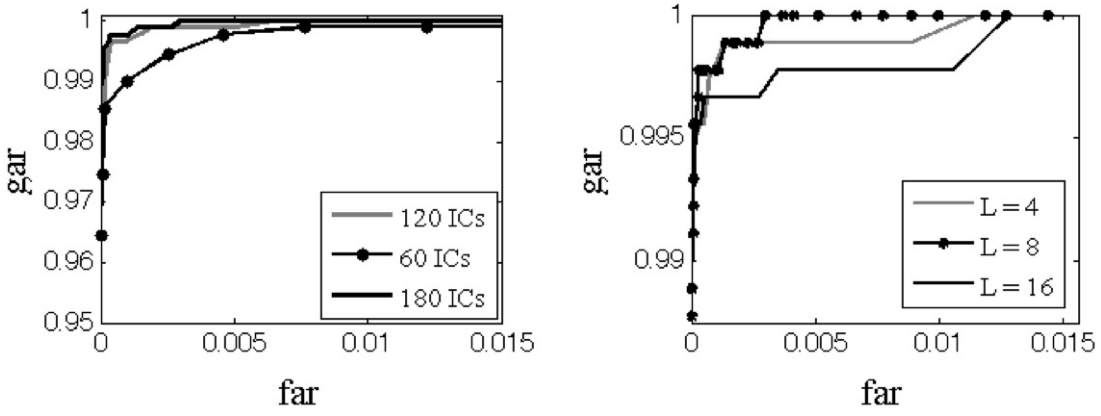
		Area			Area
L=8	60 components	0.9927	180 components	L = 4	0.9981
	120 components	0.9980		L = 8	0.9992
	180 components	0.9985		L = 16	0.9906
L=8, 180 IC	PCA	0.9944	L=8, 180 IC ICA–LE	ERICA	0.9893
	ICA	0.9992		Pearson	0.9991
	ICA–LE	0.9995		InfoMax	0.9994
				FastICA	0.9995

The area under the ROC curve is used as an index of performance in our attempt to determine a relevant configuration. An area of 0.5 indicates random performance, while an area of 1 indicates perfect performance.

Our experiments show a high performance for the proposed algorithm: the areas under the ROC curves range from 0.9927 to 0.9995 (Table 6.1). When tuning the number of ICs and quantization levels, the highest area is obtained for a high number of selected components and a medium quantization level (0.9927 vs. 0.9985 for L = 8 with 60, respectively 180 components). For a constant number of components and quantization levels, the area under the ROC curve when using ICA is larger than when using only PCA (0.9992 vs. 0.9948), but slightly lower than the area obtained by using an entropic criterion (described in chapter 4).

Examples of ROC curves for ICA architecture I are shown in Fig. 6.20–6.22. Fig. 6.20 a) compares the performance for 60, 120 and 180 components for a constant number of quantization levels. The 180 components curve is superior to the others, with a behavior close to that of a perfect classifier. In Fig. 6.20.b, the ROC curves for different quantization levels are shown. For L = 8, a 99.78% GAR and a 0.03% FAR are achieved. For a lower or a higher number of levels and the same GAR, FAR increases.

Fig. 6.21, which compares different subspace reducing strategies (described in section 4.4), illustrates the additional discrimination provided by the use of subspace selection. The entropic criterion achieves high GAR (high sensitivity) at a low FAR (high specificity).



(a) (b)

Figure 6.20: ROC curves: a) ROC curves for L = 8 quantization levels and different number of ICs; b) ROC curves for 180 components and different number of quantization levels.

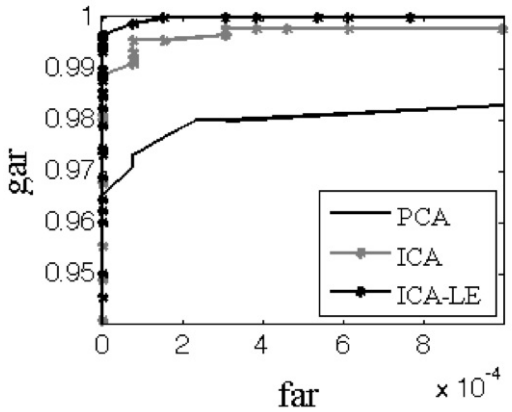


Figure 6.21: ROC curves for ICA–LE, ICA and PCA (180 ICs and 8 quantization levels).

Fig. 6.22 plots the ROC curves for various ICA algorithms (FastICA, InfoMax, Pearson, ERICA) when using an entropic criterion for subspace selection. As expected, the system's performance varies with the chosen ICA algorithm. FastICA algorithm has the highest performance in terms of ROC curve area.

Finally, this performance index leads to the following relevant configuration for the proposed methodology: FastICA algorithm with entropic criterion for subspace selection,

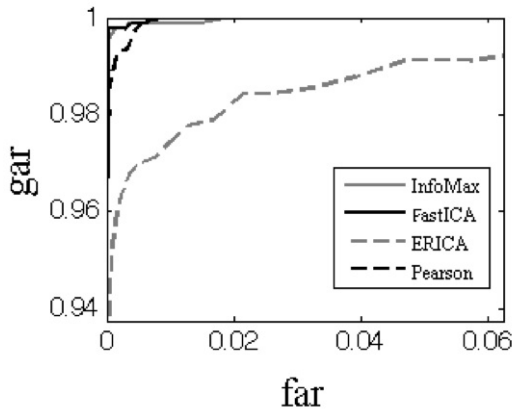


Figure 6.22: ROC curves using different ICA algorithms for 180 ICs and 8 quantization levels.

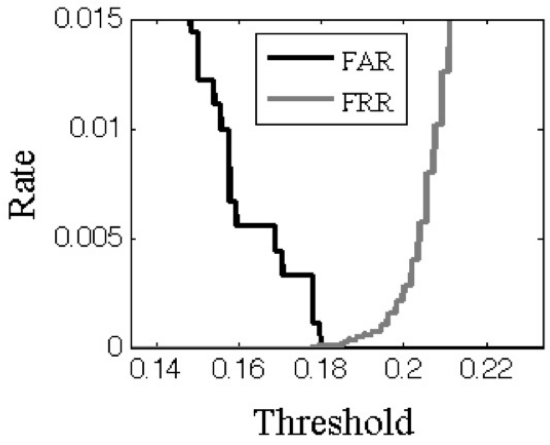


Figure 6.23: Equal error rate for 180 components and L = 8 quantization levels in ICA II approach.

with 180 ICs and $L = 8$ quantization levels.

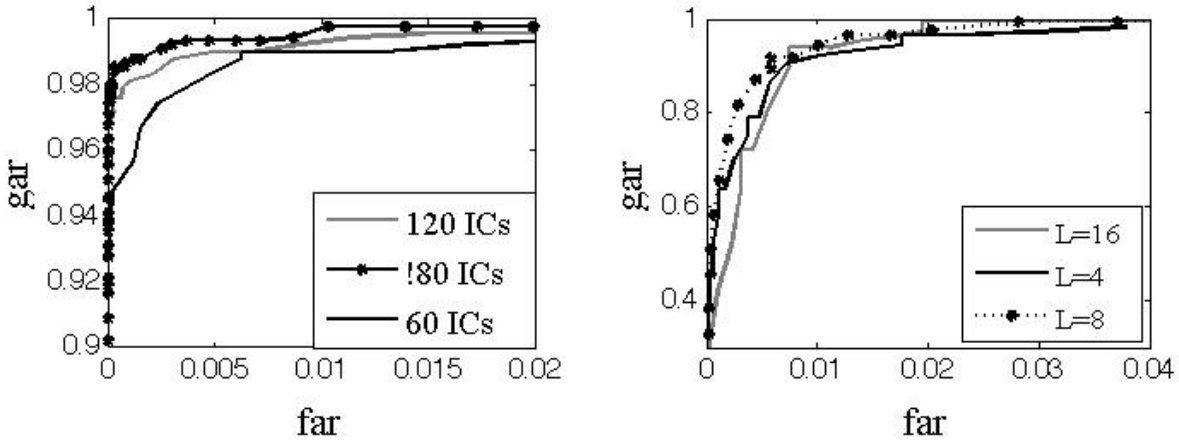
Due to the monotonically increasing nature of the ROC curve, the optimal threshold value d_0 corresponds to the maximal admissible FAR value imposed by the particular application. A FAR value of 0.007% (i.e., 7 impostors are identified as genuine among one hundred thousand persons) is common in biometric verification. For ID photograph verification, it leads to a GAR value of 99.89% (i.e. 110 genuine photographs are incorrectly identified as impostors among one hundred thousand persons) and an optimal decision threshold value of 0.17. In comparison, the equal error rate criterion (equal FAR and 1-GAR values), gives a threshold value of 0.18 (Fig. 6.23), but the associated FAR value is more than twice the admissible value (0.015%).

Examples of ROC curves for ICA architecture II are shown in Fig. 6.24–6.26. In Fig. 6.24a), the ROC curves for a different number of ICs is shown. For 180 ICs, a GAR value of 98.5% and a FAR value of 0.03% are obtained. For a lower number of ICs, for the same GAR value, FAR increases. Fig. 6.24b) presents the ROC curves for different

quantization levels. The $L = 8$ curve is superior to the other ones.

Fig. 6.25, which compares different dimension reducing strategies (PCA vs. ICAI and ICAII), illustrates the additional discrimination provided by the use of ICA. The entropic criterion achieves high sensitivity at a high specificity.

Fig. 6.26 plots the ROC curves for various image hash algorithms (ICA–LE, SVD, DWT, AlgoA). As expected, the system's performance varies with the chosen image hash algorithm. ICA–LE and AlgoA algorithm present the highest performance in terms of ROC curve area, but ICA–LE outperforms AlgoA. For ICA–LE with 180 ICs and $L = 8$, a 99.89% GAR and a 0.007% FAR is obtained. For AlgoA, for the same GAR value FAR decreases (0.03%).



(a) (b)

Figure 6.24: ROC curves for ICA approach II: a) ROC curves for L = 8 quantization levels and different number of ICs; b) ROC curves for 180 components and different number of quantization levels.

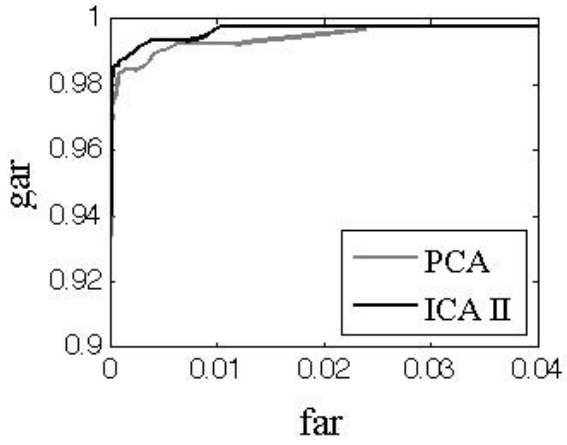


Figure 6.25: ROC curves for ICAI, ICA II and PCA for 180 ICs. The number of quantization levels for the two ICA approaches is $L = 8$.

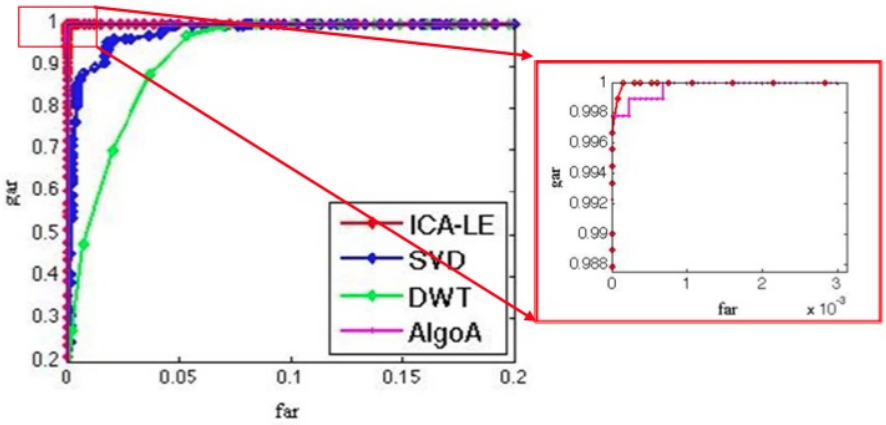


Figure 6.26: Comparison of different image hashing algorithms: ICA–LE with 180 ICs and $L = 8$, AlgoA for 100 iterations, SVD with parameters $K = 8$ and 25 rectangles of size 100, DWT with parameters $K = 8$, 150 rectangles and a Daubechies 4 wavelet for a 3–level decomposition.

6.5 Conclusion

In this chapter the developed method for printed ID image authentication was tested. The results on real data showed that the method is accurate and robust to scan attacks. The main sources of scan noise were identified and characterized. For our experimental conditions, they are CDD shot noise and, from the scanner mechanical components, jitter and reset noises.

The effects of scan noise at various stages of feature vectors extraction were analyzed by using ROC curves. The ROC analysis indicates a high accuracy in identifying the impostors. The index of performance i.e., the area under the ROC curve, varied with the number of ICs and quantization levels. The more components, the higher the accuracy. A reverse tendency was observed when increasing the number of quantization levels. The tests performed for various ICA algorithms proved that FastICA is the most suitable for our method. As expected, the highest accuracy was obtained for an entropic selection of face representation subspace. The selection by local entropy maximization allows the rejection of high global entropy components that contain most of scan noise.

Chapter 7

Conclusions and future directions

Over the past years image authentication and indexation by *hash functions* have become an important field for researchers. Such hashes are required to be content–dependent on the image and robust (the hash should be invariant under perceptually insignificant/incidental changes to the image), i.e. in applications such as database search, robustness is an important issue, facilitating image searching in large databases. An example would be locating an image that is perceptually similar to another one existing in the database, but with very different digital representation, e.g. a compressed or non–compressed image stored in a different format. Another desired property of the hash, especially in multimedia protection applications, is security. It should be impossible for an attacker to extract or reproduce the hash of a given image.

The main objective of this thesis was to propose a new technique for designing a robust image hash function for ID photos. The proposed technique can be used in ID photograph authentication systems, but it can also be used in applications such as ID photograph search.

Based on the complete analysis provided in chapter 2 of the existing image hash functions, I have decided to focus my attention, in this thesis, on dimension reduction techniques due to the promising results obtained by researchers in the past few years. The proposed method is a dimension reduction technique based on ICA decomposition.

The studied approaches propose robust image hash functions for a wide range of images, but they are not specialized on a certain image category. It is known that ID photographs must conform to some specific requirements such as single pose, neutral face expression, a minimum and maximum size of the head, etc. For this reasons, when designing a hash function for ID pictures, these characteristics should be taken into consideration.

In contrast with other dimension reduction techniques, in this thesis we propose to effect the dimension reduction in the learning stage. The first step of the method consists in pose adjustments procedure that has as main goal to center, crop and resize each image on which PCA and ICA was applied in order to obtain the projection subspace.

Next, various subspace selection techniques were introduced in order to retain the most significant components. The selection of the ICs was done by using an entropic criterion. By projecting the images only on the significant ICs, the robustness of the

method to non–malicious attacks such as print–and–scan was increased.

Important contributions were presented in chapter 6 where the effects of scan noise were analyzed. The main sources of scan noise were identified and characterized. For the given experimental conditions, the CCD shot noise, the jitter and the reset noise were identified as the most significant scan noises. The effect of these noises was analyzed at various stages of the feature vectors extraction.

Personal contributions

The contributions of the thesis can be summarized as follows:

- I proposed and implemented a no–key learning based method for generating the intermediate hash for ID photographic images. The software was implemented in Matlab;
- I showed the importance of components selection in the learning stage. I developed two algorithms based on an entropic criterion of components selection that improve the results compared to the case when no subspace selection is done;
- For experiments I created the image database from the FERET database by selecting only the faces with neutral and frontal view.
- I tested the method on simulated and real data;
- I studied the noise introduced by the flat–bed scanner employed for the experiments. I identified the main sources of noise and I developed a noise model for the studied scanner by taking into account these sources;
- I highlighted the fact that the hashing algorithms developed for general images are not very performant in the particular case of ID photograph images;
- Using this algorithm an image authentication system can be implemented. The system consists in a learning and a verification stage as presented in chapter 4. The system should offer good performance in terms of speed and discriminability.

Future work

Further researches can be carried out in different directions:

- improving the method to print–and–scan attacks by completing the proposed scheme with a decoding stage that will ensure further robustness and also security. Frequently used decoding algorithms such as Reed–Muller [Lin 2005] or clustering algorithm could be employed in order to obtain the final hash value. The clustering algorithms seem more adequate for final hash value generation and can be improved in order to increase the performance of the method;

- enforcing the security of the method by developing a key–based version of this algorithm;
- testing the proposed method also in the case of image forgery. It is interesting to see if the method is robust to malicious attacks such as image content modification made by an unauthorized person;
- testing all the print–and–scan models presented in chapter 5 on the existing image database and realize a complete analysis of the obtained results in order to see which algorithm simulates at best the print–and–scan process;
- developing a new print–and–scan chain model based on the existing statical models in the literature and on the scan noise model proposed in this thesis. It is desirable to have a simple model which simulates at best the print–and–scan effects on an image.

Appendix A

Appendix A

	Attack	r-value	Attack	r-value
180 PCA	AFFINE _1	3.95	AFFINE _2	1.4
$L = 8$	AFFINE _3	3.78	AFFINE _4	1.5
	AFFINE _5	3.28	AFFINE _6	2.98
	AFFINE _7	3.12	AFFINE _8	3.1
	WN $\sigma = 0.01$	2.5	WN $\sigma = 0.02$	1.9
	Median 3×3	3.63	Median 5×5	2.55
120 PCA	AFFINE _1	3.1	AFFINE _2	1.36
$L = 8$	AFFINE _3	3.3	AFFINE _4	1.4
	AFFINE _5	2.8	AFFINE _6	2.6
	AFFINE _7	2.7	AFFINE _8	2.6
	WN $\sigma = 0.01$	2.3	WN $\sigma = 0.02$	2.3
	Median 3×3	3.1	Median 5×5	2.4
120 PCA	AFFINE _1	2.7	AFFINE _2	1.3
$L = 4$	AFFINE _3	2.7	AFFINE _4	1.3
	AFFINE _5	2.4	AFFINE _6	2.2
	AFFINE _7	2.3	AFFINE _8	2.2
	WN $\sigma = 0.01$	2	WN $\sigma = 0.02$	1.7
	Median 3×3	2.6	Median 5×5	2.2
120 LE	AFFINE _1	3.8	AFFINE _2	1.4
$L = 8$	AFFINE _3	4	AFFINE _4	1.8
	AFFINE _5	3.5	AFFINE _6	3.2
	AFFINE _7	3.3	AFFINE _8	3.2
	WN $\sigma = 0.01$	2.5	WN $\sigma = 0.02$	1.9
	Median 3×3	3.8	Median 5×5	2.8
	AFFINE _1	4.8	AFFINE _2	1.7

Table A.1: Results for different attacks and selection criteria for InfoMax algorithm: PCA and LE.

	Attack	r–value	Attack	r–value
180 PCA	AFFINE _1	3.5	AFFINE _2	1.2
$L = 8$	AFFINE _3	4.1	AFFINE _4	1.6
	AFFINE _5	3.2	AFFINE _6	2.8
	AFFINE _7	3	AFFINE _8	2.5
	WN $\sigma = 0.01$	99.1	WN $\sigma = 0.02$	1.9
	Median 3×3	4	Median 5×5	2.9
120 PCA	AFFINE _1	3	AFFINE _2	1
$L = 8$	AFFINE _3	3.9	AFFINE _4	1.5
	AFFINE _5	2.9	AFFINE _6	2.5
	AFFINE _7	2.6	AFFINE _8	2.1
	WN $\sigma = 0.01$	2.3	WN $\sigma = 0.02$	2.3
	Median 3×3	3.8	Median 5×5	2.8
120 PCA	AFFINE _1	2.6	AFFINE _2	1
$L = 4$	AFFINE _3	3.3	AFFINE _4	1.4
	AFFINE _5	2.5	AFFINE _6	2.2
	AFFINE _7	2.3	AFFINE _8	2
	WN $\sigma = 0.01$	2	WN $\sigma = 0.02$	1.7
	Median 3×3	3.1	Median 5×5	2.4
120 LE	AFFINE _1	3.5	AFFINE _2	1.2
$L = 8$	AFFINE _3	4.3	AFFINE _4	1.3
	AFFINE _5	3.1	AFFINE _6	2.81
	AFFINE _7	3	AFFINE _8	2.5
	WN $\sigma = 0.01$	2.5	WN $\sigma = 0.02$	1.9
	Median 3×3	4.2	Median 5×5	3

Table A.2: Results for different attacks and selection criteria for ERICA algorithm: PCA and LE.

	Attack	r–value	Attack	r–value
180 PCA	AFFINE _1	3.7	AFFINE _2	1.7
$L = 8$	AFFINE _3	3.7	AFFINE _4	1.7
	AFFINE _5	3.2	AFFINE _6	3
	AFFINE _7	3.1	AFFINE _8	3
	WN $\sigma = 0.01$	2.5	WN $\sigma = 0.02$	1.9
	Median 3×3	3.4	Median 5×5	2.6
120 PCA	AFFINE _1	3.7	AFFINE _2	1.4
$L = 8$	AFFINE _3	3.8	AFFINE _4	1.5
	AFFINE _5	3.2	AFFINE _6	2.9
	AFFINE _7	3.1	AFFINE _8	3
	WN $\sigma = 0.01$	2.3	WN $\sigma = 0.02$	2.3
	Median 3×3	3.4	Median 5×5	2.5
120 PCA	AFFINE _1	3.1	AFFINE _2	1.4
$L = 4$	AFFINE _3	3.3	AFFINE _4	1.5
	AFFINE _5	2.8	AFFINE _6	2.6
	AFFINE _7	2.8	AFFINE _8	2.6
	WN $\sigma = 0.01$	2	WN $\sigma = 0.02$	1.7
	Median 3×3	3	Median 5×5	2.4
120 LE	AFFINE _1	3.9	AFFINE _2	1.5
$L = 8$	AFFINE _3	3.7	AFFINE _4	1.3
	AFFINE _5	3.3	AFFINE _6	2.8
	AFFINE _7	3.1	AFFINE _8	2.9
	WN $\sigma = 0.01$	2.5	WN $\sigma = 0.02$	1.9
	Median 3×3	3.5	Median 5×5	2.4

Table A.3: Results for different attacks and selection criteria for Pearson algorithm: PCA and LE.

Acronyms

ID identity document 1
ICA Independent Component Analysis vii
PCA Principal Component Analysis 2
MD5 Message Digest 5 5
SHA Secure Hash Algorithm 5
LBG Linde–Buzo–Gray 12
DCT Discrete Cosine Transform 13
WT Wavelet Transform 12
DWT Discrete Wavelet Transform 14
SDS structural digital signature 14
RSA Rivest, Shamir, Adleman 15
VLC Variable Length Code 20
SVD Singular Value Decomposition 21
PR pseudo–random 21
NMF nonnegative matrix factorization 22
FJLT Fast Johnson–Lindenstrauss Transform 23
RI–FJLT rotation invariant Fast Johnson–Lindenstrauss Transform 24
MT Mellin Transform 24
FFT Fast Fourier Transform 27
PC Principal Component 44
ATM automated teller machine 73
RIP Raster Image Processor 75
CCD charged–couple device 76
CIS contact image sensor 76

ADC analog–to–digital converter 76

RST rotation, scaling, translation 78

RSC rotation, scaling, cropping 82

GUI Graphical User Interface 82

GGD Generalized Gaussian Distribution 37

IC Independent Component x

LE local entropy 55

ROC Receiver Operating Characteristic 54

FAR false acceptance rate 60

GAR genuine acceptance rate 61

FRR false rejection rate 60

FastICA Fast Independent Component Analysis 32

EFICA Efficient Fast Independent Component Analysis 37

List of Scientific Publications

CONFERENCE PAPERS

1. A. Smoacă, J. M. Becker, M. Goeb, D. Colţuc, *An Efficient Algotithm for a New Circularity Assessment Measure*, Stereology and Image Analysis. Ecs10: Proceedings of the 10th European Conference of ISS., (V.Capasso et al. Ed.), The MIRIAM Project Series, Vol. 4, ESCULAPIO Pub. Co., Bologna, Italy, 2009. ISBN: 978-88-7488-310-3.

2. A. Smoacă, D. Colţuc, V. Lăzărescu, *Pattern Segmentation in Textile Images*, Proceedings of ISSCS 2009, Iasi, Romania, ISBN: 978-1-4244-3785-6, pg. 137-140.

3. A. Smoacă, M. Petrovici, D. Colţuc, V. Lăzărescu, *A Robust Hashing of ID Photos*, International Symposium on Signals, Circuits and systems , ISSCS Iasi, ISBN 978-1-4577-0201-3, pp. 43-46, 2011.

4. A. Smoacă, D. Colţuc and T. Fournel, *Image Characterization by Entropic Biometric Decomposition*, AIP Conference Proceedings, ISSN: 0094-243X, vol. 1305, no. 1, pages 381–388, 2011.

SUBMITTED PAPERS

1. A. Smoacă, D. Colţuc and T. Fournel, *Authentication of printed ID photographs via local ICA features*, Journal of Visual Communication and Image Representation (ISI).

2. A. Smoacă, V. Lăzărescu, *ID image decomposition using two different ICA approaches*, UPB Journal.

Bibliography

[Amiri 2009] S. Amiri and Mansour Jamzad. *An Algorithm for Modeling Print and Scan Operations Used for Watermarking.* In Digital Watermarking, volume 5450 of *Lecture Notes in Computer Science*, pages 254–265. Springer Berlin/Heidelberg, 2009. (Cited on pages 81 and 86.)

[Aoki 1998] S. Aoki. *New halftoning method using adaptive cell.* In Proceedings of the IS&T's NIP 14: International Conference on Digital Printing Technologies, pages 277–280, 1998. (Cited on page 98.)

[Baier] S. Baier. *CCD Imaging Systems.* Application Note. (Cited on page 98.)

[Bartlett 1998] Marian Stewart Bartlett and H. Martin Lades. *Independent Component Representations for Face Recognition.* In Storage and Retrieval for Image and Video Databases, 1998. (Cited on pages 43 and 44.)

[Bartlett 2001] M.S. Bartlett. Face image analysis by unsupervised learning. Kluwer Academic, Dordrecht, 2001. (Cited on page 41.)

[Bartlett 2002] M.S. Bartlett, J.R. Movellan and T.J. Sejnowski. *Face Recognition by Independent Component Analysis.* IEEE Trans. Neural Networks, vol. 13, no. 6, pages 1450–1464, 2002. (Cited on pages 41, 43 and 44.)

[Bell 1995] Anthony J. Bell and Terrence J. Sejnowski. *An information-maximization approach to blind separation and blind deconvolution.* NEURAL COMPUTATION, vol. 7, pages 1129–1159, 1995. (Cited on pages 33 and 37.)

[Bhattacharjee 1998] Sushil Bhattacharjee and Martin Kutter. *Compression Tolerant Image Authentication.* In Proc. IEEE Conf. on Image Processing, pages 435–439, 1998. (Cited on pages 10 and 19.)

[Blahut 1994] R. E. Blahut. Theory and practice of error-correcting codes. Addison Wesley, 1st edition, 1994. (Cited on page 13.)

[Campbell 1996] F. W. Campbell, J. J. Kulikowski and J. Levinson. *The effect of orientation on the visual resolution of gratings.* The Journal of Physiology, vol. 187, pages 427–436, 1996. (Cited on page 95.)

[Cardoso 1997] J. F. Cardoso. *Infomax and maximum likelihood for source separation.* IEEE Letters on Signal Processing, vol. 4, pages 112–114, 1997. (Cited on pages 33 and 34.)

[Comon 1994] Pierre Comon. *Independent component analysis, a new concept?* Signal Process., vol. 36, pages 287–314, April 1994. (Cited on page 34.)

[Coudray 1996] M. A. Coudray. *Causes and corrections of dot gain on press.* Screen Printing: The Journal of Technology and Management, vol. 86, no. 9, pages 18–26, 1996. (Cited on page 91.)

[Cox 1997] Ingemar J. Cox, Joe Kilian, F. Thomson Leighton and Talal Shamoon. *Secure spread spectrum watermarking for multimedia.* IEEE Transactions on Image Processing, vol. 6, no. 12, pages 1673–1687, 1997. (Cited on page 6.)

[Cox 2002] Ingemar Cox, Matthew L. Miller and Jeffery A. Bloom. Digital watermarking. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2002. (Cited on page 5.)

[Daugman 1993] J. Daugman. *High Confidence Cisual Recognition of Persons by a Test of Statistical Independence.* IEEE Trans. Pattern Anal. Mach. Intell., vol. 15, no. 11, pages 1148–1161, 1993. (Cited on page 106.)

[de Lathauwer 1995] L. de Lathauwer, P. Comon, B. de Moor and J. Vandewalle. *Higher-order power method–application in Independent Component Analysis.* In Proc. of the International Symposium on Nonlinear Theory and its Applications (NOLTA'95), pages 91–96, 1995. (Cited on page 34.)

[Degara-Quintela 2003] Norberto Degara-Quintela and Fernando Perez-Gonzalez. *Visible encryption: using paper as a secure channel.* In Proc. SPIE, volume 5020, pages 413–422, 2003. (Cited on pages xi, 81, 85 and 86.)

[Dittmann 1999] Jana Dittmann, Arnd Steinmetz and Ralf Steinmetz. *Content-Based Digital Signature for Motion Pictures Authentication and Content-Fragile Watermarking.* In Proceedings of the IEEE International Conference on Multimedia Computing and Systems - Volume 2, ICMCS '99, pages 209–, Washington, DC, USA, 1999. IEEE Computer Society. (Cited on pages 10 and 19.)

[Doran 2007] Robert W. Doran. *The Gray Code.* Journal of Universal Computer Science, vol. 13, no. 11, pages 1573–1597, nov 2007. (Cited on page 49.)

[Ekdemir 2011] S. Ekdemir and X. Wu. *Digital Halftoning : Improvements on the Two-by-Two Block Replacement Method.* Technical report, Uppsala Universitet, 2011. (Cited on pages 95 and 96.)

[Fawcett 2006] T. Fawcett. *An introduction to ROC analysis.* Pattern Recognition Letters, vol. 27, no. 8, 2006. (Cited on page 61.)

[Fridrich 2000] Jiri Fridrich and M. Goljan. *Robust Hash Functions for Digital Watermarking.* International Conference on Information Technology: Coding and Computing, vol. 0, page 178, 2000. (Cited on pages 8, 10 and 17.)

[Gray 1953] F. Gray. *Pulse Code Communication.* U. S. Patent 2632058, 1953. (Cited on page 49.)

[Haas 2009] Bertrand Haas and Robert A. Cordery. *Method and system for optimizing print-scan simulations.* Patent 7586627, 2009. (Cited on page 89.)

[Haykin 1998] S. Haykin. Neural networks – a comprehensive foundation. Prentice Hall, 2nd edition, 1998. (Cited on page 31.)

[He 2004] Zhen He and Charles A. Boumani. *AM/FM halftoning: digital halftoning through simultaneous modulation of dot size and dot density.* J. Electron. Imaging, vol. 13, no. 2, pages 286–302, 2004. (Cited on pages 95 and 96.)

[Herault 1984] J. Herault and B. Ans. *Circuits neuronaux à synapses modifiables: décodage de messages composites par apprentissage non supervise.* C.R. de l'Academie des Sciences, vol. 229, pages 525–528, 1984. (Cited on page 34.)

[Holliman 1999] M. Holliman, N. Memon and M. M. Yeung. *On the Need for Image Dependent Keys for Watermarking.* In Proceedings of Content Security and Data Hiding in Digital Media, 1999. (Cited on page 7.)

[Hyvärinen 1997] Aapo Hyvärinen. *One–Unit Contrast Functions For Independent Component Analysis: A Statistical Analysis.* In Proceedings of the 1997 IEEE Workshop Neural Networks for Signal Processing [1997] VII, pages 388 – 397, 1997. (Cited on pages 32 and 35.)

[Hyvärinen 1999] A. Hyvärinen. *Fast and Robust Fixed-Point Algorithms for Independent Component Analysis.* IEEE Transactions on Neural Networks, vol. 10, pages 626–634, 1999. (Cited on page 36.)

[Hyvärinen 2001] A. Hyvärinen, J. Karhunen and E. Oja. Pattern classification. John Wiley & Sons, 2001. (Cited on pages 29, 31, 32, 34 and 35.)

[Kailasanathan 2001] C. Kailasanathan, R. Safavi-Naini and P. Ogunbona. *Image Authentication Surviving Acceptable Modifications.* In Proc. IEEE-EURASIP Workshop on Nonlinear Signal Image Processing, Baltimore, MD, 2001. (Cited on pages 6, 10 and 11.)

[Karhunen 1997] J. Karhunen, E. Oja, L. Wang, R. Vigario and J. Joutsensalo. *A class of neural network for Independent Component Analysis.* IEEE Transactions on Neural Networks, vol. 8, pages 2486 – 504, 1997. (Cited on page 36.)

[Karvanen 2000] J. Karvanen, J. Eriksson and V. Koivunen. *Pearson System Based Method for Blind Separation.* In Proceedings of Second International Workshop on Independent Component Analysis and Blind Signal Separation, 2000. (Cited on page 38.)

[Kodak 2005] Kodak. *CCD Image Sensor Noise Sources.* Application Note, 2005. (Cited on page 98.)

[Koldovsky 2006] Z. Koldovsky, P. Tichavsky and E. Oja. *Fast and Robust Fixed-Point Algorithms for Independent Component Analysis.* IEEE Transactions on Neural Networks, vol. 17, pages 1265 – 1277, 2006. (Cited on pages ix and 37.)

[Kozat 2004] S. S. Kozat, K. Mihcak and R. Venkatesan. *Robust perceptual image hashing via matrix invariances.* In Proc. IEEE Conf. on Image Processing, pages 3443–3446, 2004. (Cited on pages 10, 21 and 22.)

[Kundu 2006] M.K. Kundu and A.K. Maiti. *An inexpensive digital watermarking scheme for printed document.* IET Conference Publications, vol. 2006, no. CP522, pages 378–383, 2006. (Cited on pages 81 and 86.)

[Lau 1998] D. L. Lau, G. R. Arce and N. C. Gallagher. *Green-noise digital halftoning.* In Proceedings of the IEEE, volume 86, pages 2424–2444, 1998. (Cited on page 98.)

[Lau 2008] D.L. Lau and G.R. Arce. Modern digital halftoning. CRC Press, Taylor and Francis Croup, second edition, 2008. (Cited on page 91.)

[Lee 2001] Daniel D. Lee and H. Sebastian Seung. *Algorithms for Non-negative Matrix Factorization.* In NIPS, pages 556–562. MIT Press, 2001. (Cited on page 22.)

[Lin 1997] Ching-Yung Lin and Shih fu Chang. *A Robust Image Authentication Method Distinguishing JPEG Compression from Malicious Manipulation.* IEEE Transactions on Circuits and Systems of Video Technology, vol. 11, pages 153–168, 1997. (Cited on pages ix, 10, 13 and 15.)

[Lin 1999a] Ching-Yung Lin and Shih-Fu Chang. *Distortion Modeling and Invariant Extraction for Digital Image Print-and-Scan Process.* In International Symposium on Multimedia Information Processing (ISMIP 99), Taipei, Taiwan, 1999. (Cited on page 73.)

[Lin 1999b] Eugene T. Lin and Edward J. Delp. *A Review of Fragile Image Watermarks.* In Proc. ACM Multimedia and Security Workshop, volume 1, pages 25–29, 1999. (Cited on pages 6, 54, 81, 82 and 83.)

[Lin 2001] Shu Lin. *An Extendible Hash for Multi-Precision Similarity Querying of Image Databases.* In Proc. of the 27th VLDB Conference, 2001, pages 221–230, 2001. (Cited on page 7.)

[Lin 2005] Shu Lin and Daniel Costello. Error control coding. Pearson, 2nd edition, 2005. (Cited on page 114.)

[Liu 1999] C. Liu and H. Wechsler. *Comparative Assessment of Independent Component Analysis (ICA) for Face Recognition.* In Proc. of the 2nd International Conference on Audio- and Video-based Biometric Person Authentication, 1999. (Cited on page 44.)

[Lu 2000] Chun-Shien Lu and Hong-Yuan Mark Liao. *Structural digital signature for image authentication: an incidental distortion resistant scheme.* In Proceedings of the 2000 ACM workshops on Multimedia, MULTIMEDIA '00, pages 115–118, New York, NY, USA, 2000. ACM. (Cited on pages ix, 10, 14 and 16.)

[Lv 2009] Xudong Lv and Z. Jane Wang. *An extended image hashing concept: content-based fingerprinting using FJLT.* EURASIP Journal on Information Security, vol. 2009, pages 2:1–2:16, January 2009. (Cited on page 23.)

[Malvido 2006] Alberto Malvido, O. Pérez-González and O. Cousino. *A Novel Model for the Print–and–Capture Channel in 2D Bar Code.* In International Workshop on Multimedia Content Representation, Clasiffication and Security, LNCS 4105, pages 627–634. Springer-Verlag, 2006. (Cited on pages 81 and 85.)

[Martinian 2002] Emin Martinian and Gregory W. Wornell. *Multimedia Content Authentication: Fundamental Limits.* In Proc. IEEE Int. Conf. Image Processing, Rochester, NY, 2002. (Cited on page 6.)

[Menezes 1996] Alfred J. Menezes, Scott A. Vanstone and Paul C. Van Oorschot. Handbook of applied cryptography. CRC Press, Inc., Boca Raton, FL, USA, 1st edition, 1996. (Cited on page 16.)

[Mihçak 2001] M. Kivanç Mihçak and Ramarathnam Venkatesan. *A Perceptual Audio Hashing Algorithm: A Tool For Robust Audio Identification . . .* In Proceedings of 4th Information Hiding Workshop, pages 51–65, 2001. (Cited on pages 6, 10, 17 and 20.)

[Monga 2005] V. Monga, D. Vats and B.L. Evans. *Image Authentication Under Geometric Attacks Via Structure Matching.* In IEEE International Conference on Multimedia and Expo, 2005. ICME 2005, pages 229–232, 2005. (Cited on page 26.)

[Monga 2006] V. Monga and B.L. Evans. *Perceptual Image Hashing Via Feature Points: Performance Evaluation and Tradeoffs.* IEEE Transactions on Image Processing, vol. 15, pages 3452–3465, 2006. (Cited on pages 20, 21 and 24.)

[Monga 2007] V. Monga and B.L. Evans. *Perceptual Image Hashing Via Feature Points: Performance Evaluation and Tradeoffs.* IEEE Transactions on Information Forensics and Security, vol. 2, pages 376 – 390, 2007. (Cited on page 22.)

[NIST 2008] NIST. Secure hash standard (shs) [electronic resource]. U.S. Dept. of Commerce, National Institute of Standards and Technology, 2008. (Cited on page 5.)

[Pearson 1901] K. Pearson. *On lines and planes of closest fit to systems of points in space*. Philosophical Magazine, vol. 2, pages 559–572, 1901. (Cited on page 29.)

[Petitcolas 1998] Fabien A. P. Petitcolas, Ross J. Anderson and Markus G. Kuhn. *Attacks on copyright marking systems*. In Proceedings of the Information Hiding, Second International Workshop, IH'98, LNCS 1525, pages 219–239. Springer Verlag, 1998. (Cited on page 54.)

[Petitcolas 2000] Fabien A. P. Petitcolas. *Watermarking schemes evaluation*. IEEE Signal Processing, vol. 17, no. 5, pages 58–64, September 2000. (Cited on page 54.)

[Pham 1992] D.T. Pham, P. Garrat and C. Jutten. *Separation of a mixture of independent sources through a maximum likelihood approach*. In Proc. EUSIPCO, pages 771–774, 1992. (Cited on page 33.)

[Phillips 1998] P. Jonathon Phillips, Wechsler H., Huang J. and Patrick J. Rauss. *The FERET database and evaluation procedure for face-recognition algorithms*. Image and Vision Computing, vol. 16, no. 5, pages 295–306, 1998. (Cited on page 53.)

[Phillips 2000] P. Jonathon Phillips, Hyeonjoon Moon, Syed A. Rizvi and Patrick J. Rauss. *The FERET Evaluation Methodology for Face-Recognition Algorithms*. IEEE Trans. Pattern Anal. Mach. Intell., pages 1090–1104, 2000. (Cited on page 53.)

[Rivest 1992] R. Rivest. *The MD5 Message-Digest Algorithm*. 1992. (Cited on page 5.)

[Schneider 1996] Marc Schneider and Shih fu Chang. *A Robust Content Based Digital Signature For Image Authentication*. In Proc. IEEE Conf. on Image Processing, volume 3, pages 227–230, 1996. (Cited on page 10.)

[Smoaca 2011] A. Smoaca, D. Coltuc and T. Fournel. *ID Image Characterization by Entropic Biometric Decomposition*. AIP Conference Proceedings, vol. 1305, no. 1, pages 381–388, 2011. (Cited on pages 45 and 50.)

[Solanki 2005] Kaushal Solanki, Upamanyu Madhow, Banglore S. Manjunath and Shiv Ch. *Modeling the Print-Scan Process for Resilient Data Hiding*. In Proc. SPIE, pages 418–429, 2005. (Cited on pages 80 and 98.)

[Solanki 2006] K. Solanki, U. Madhow, B. S. Manjunath, S. Chandrasekaran and I. El-Khalil. *'Print and Scan' Resilient Data Hiding in Images*. IEEE Transactions on Information Forensics and Security, vol. 1, no. 4, pages 464–478, Dec 2006. (Cited on page 77.)

[Su 2005] Karen Su, Deepa Kundur and Dimitrios Hatzinakos. *Statistical Invisibility for Collusion-resistant Digital Video Watermarking*. IEEE Trans. Multimedia, vol. 7, pages 43–51, 2005. (Cited on page 7.)

[Swaminathan 2002] A. Swaminathan, Yinian Mao and Min Wu. *Robust and secure image hashing*. IEEE Transactions on Information Forensics and Security, vol. 1, pages 215–230, 2002. (Cited on page 24.)

[Taylor 1998] Stuart A. Taylor. *CCD and CMOS Imaging Array Technologies: Technology Review*. Technical report, 1998. (Cited on page 76.)

[Tong 1991] L. Tong, R.-w. Liu, V.C. Soon and Y.-F. Huang. *Indeterminacy and identifiability of blind identification*. Circuits and Systems, IEEE Transactions on, vol. 38, no. 5, pages 499 –509, May 1991. (Cited on page 35.)

[Tsalaile 2009] T. Tsalaile, R. Sameni, S. Sanei, C. Jutten and J. Chambers. *Sequential Blind Source Extraction For Quasi-Periodic Signals With Time-Varying Period*. IEEE Transactions on Biomedical Engineering, vol. 56, no. 3, pages 646 –655, March 2009. (Cited on page 35.)

[Venkatesan 2000] R. Venkatesan, S. m. Koon, M. H. Jakubowski and P. Moulin. *Robust Image Hashing*. In Proc. IEEE Int. Conf. Image Processing, Vancouver, BC, Canada, volume 2006, pages 664–666, 2000. (Cited on pages 6, 7, 10 and 12.)

[Vikas 2005] R. Vikas and K.K. Barman. *A report on Print-Scan Resilient Information Hiding In Images*. Technical report, Hewlett-Packard Labs, 2005. (Cited on page 90.)

[Villán 2005] Renato Villán, Sviatoslav Voloshynovskiy, Oleksiy Koval and Thierry Pun. *Multilevel 2D Bar Codes: Towards High Capacity Storage Modules for Multimedia Security and Management*. In Proceedings of SPIE-IS&T Electronic Imaging 2005, Security, Steganography, and Watermarking of Multimedia Contents VII, pages 453–464, 2005. (Cited on pages xi, 81, 83, 84, 85 and 86.)

[Voloshynovskiy 2004] Sviatoslav Voloshynovskiy, Oleksiy Koval, Frederic Deguillaume and Thierry Pun. *Visual communications with side information via distributed printing channels: extended multimedia and security perspectives*. In Security, Steganography, and Watermarking of Multimedia Contents, pages 428–445, 2004. (Cited on page 81.)

[Wolfgang 1999] Raymond B. Wolfgang and Edward J. Delp. *Fragile Watermarking Using the VW2D Watermark*. In Proc. SPIE/IS&T Inter. Conf. Security and Watermarking of Multimedia Contents, pages 204–213, 1999. (Cited on page 6.)

[Wu 1998] Min Wu and Bede Liu. *Watermarking for Image Authentication*. In Proc. IEEE Conf. on Image Processing, volume 2, pages 437–441, 1998. (Cited on page 6.)

[Wu 2002] Min Wu. *Multimedia Data Hiding*. In Digital Binary Image, IEEE Int. Conf. Multimedia and Expo, ICME'00. Springer–Verlag, 2002. (Cited on page 5.)

[Wu 2009] Di Wu, Xuebing Zhou and Xiamu Niu. *A Novel Image Hash Algorithm Resistant to Print–scan*. Signal Process, vol. 89, pages 2415–2424, December 2009. (Cited on page 26.)

[www.howstuffworks.com] www.howstuffworks.com. *http://www.howstuffworks.com/*. (Cited on pages x, 74 and 75.)

[Xie 2001] L. Xie and G. R. Arce. *A class of authentication digital watermarks for secure multimedia communication*. IEEE Trans. on Image Processing, vol. 10, pages 1754–1764, 2001. (Cited on page 6.)

[Yeung 1997] Minerva M. Yeung and Fred Mintzer. *An Invisible Watermarking Technique for Image Verification*. International Conference on Image Processing, vol. 2, page 680, 1997. (Cited on page 6.)

[Yu 2005] L. Yu, X. Niu and S. Sun. *Print-and-scan model and the watermarking countermeasure*. Image Vision Comput., vol. 23, pages 807–814, September 2005. (Cited on pages 79 and 86.)

[Yu 2007] Longjiang Yu and Shenghe Sun. *Image Authentication in Print-and-scan Scenario*. In Proceedings of the Third International Conference on International Information Hiding and Multimedia Signal Processing (IIH-MSP 2007) - Volume 01, IIH-MSP '07, pages 295–298, Washington, DC, USA, 2007. IEEE Computer Society. (Cited on pages 26 and 105.)

[Zhang 2009] Yongping Zhang, Xiangui Kang and Philipp Zhang. *A Practical Print-and-Scan Resilient Watermarking for High Resolution Images*. In Hyoung-Joong Kim, Stefan Katzenbeisser and Anthony Ho, editeurs, Digital Watermarking, volume 5450 of *Lecture Notes in Computer Science*, pages 103–112. Springer, 2009. (Cited on page 81.)

[Zitova 1999] Barbara Zitova, Jaroslav Kautsky, Gabriele Peters and Jan Flusser. *Robust Detection of Significant Points in Multiframe Images*. Pattern Recognition Letters, vol. 20, pages 199–206, 1999. (Cited on page 26.)